REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1997	Annual/Final, Aug 1995-Jan 1997

4. TITLE AND SUBTITLE
New Models and Fast Algorithms for
Natural and Urban Clutter with Applications

5. FUNDING NUMBERS
F49620-95-1-0499
AFOSR-TR
97-0680

6. AUTHOR(S)
Dr. David B. Cooper

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Brown University, Providence, RI 02912

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
110 Duncan Avenue, Suite B115
Bolling AFB, DC 20332

nm

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

To address the Automatic Target Detection/Recognition (ATD/R) community's long term goal of "revolutionizing wide-area imagery analysis", the research presented in this report focused on two enabling technologies: (1) clutter models based on Synthetic Aperture Radar (SAR) in urban environments; and (2) fast algorithms for Markov Random Fields. The former investigated the nature of building signatures in SAR imagery, and saw the development of a building detector. Buildings contribute a large number of false target detections, and targets stationed in close proximity to buildings can be missed using conventional analysis methods. The latter effort revisited signal processing to make theoretical headway in developing reduced computational cost techniques for estimating and using stochastic target and clutter models based on Markov Random Fields. These algorithms have broad application to a large variety of ATD/R concerns.

14. SUBJECT TERMS	15. NUMBER OF PAGES
SAR, building detection, Markov Random Fields	33
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Contents

1 Executive Summary

This report summarizes the technical details of our work under the University Research Initiative (URI) effort in Automatic Target Detection/Recognition (ATD/R). At the outset of our work, a long-term goal of great concern for the ATD/R community was "revolutionizing wide-area imagery analysis". We structured our work with this ideal in mind, focusing on two enabling technologies: (1) the design of clutter models based on Synthetic Aperture Radar (SAR) in urban environments; and (2) fast algorithms for Markov Random Fields. In addition, we engaged in a small, yet rewarding, effort in which we developed a novel method for fitting implicit algebraic curves and surfaces to two- and three-dimensional data. The remainder of this section is devoted to an overview of these research areas. The technical details of our research are discussed in Sec. 2.

1.1 SAR Building Detection in Urban Environments

Because SAR is an all-weather sensor, and is operable both night and day, it can be used in situations (cloud cover, smoke/fog, darkness) where Electro-Optical (EO) sensors are practically useless. This makes SAR an extremely useful tool for surveillance. However, since radar derived imagery was historically of low resolution, sophisticated object detection/recognition schemes, like those used to process EO images, were not available. Recent developments in sensor technology have made it possible to obtain multi-channel, high-resolution SAR imagery. We apply sophisticated machine vision algorithms to such imagery to analyze and detect objects of interest.

As a modality of choice for wide-area surveillance, SAR has enjoyed a good deal of success detecting targets of interest in natural settings. However, when employed in urban environments, SAR performance simply is not comparable. Here, the presence of cultural, non-target but *target-like* objects - buildings, vehicles, bridges - greatly increases the number of false alarms the SAR system reports. Furthermore, the SAR signature of a target can be greatly distorted as a function of the target's physical location with respect to a non-target object. That is, the SAR signature of a tank positioned in front of the leading edge of a building might be very unlike the SAR signature of a tank alone. This is because some reflections from the tank will have also reflected off the building wall. In addition, returns from the building alone will arrive roughly coincidentally with the reflections from the tank alone. Successful cultural object model development, enabling the detection of a variety of cultural objects of interest, would contribute to the goals of MSTARS, RADIUS, and other Monitor programs, and might influence others, such as ARPA's Clipping Service.

Ultimately, it is necessary to have SAR based models for the complete urban environment. For this effort, our focus centered on buildings, in particular, models for detection and two-dimensional geometry estimation. Detecting buildings, as well as estimating building geometry and appearance in SAR imagery, is of interest for two important reasons: (1) Buildings contribute a large number of false alarms. By detecting buildings, the false alarm rate can be greatly reduced; and (2) Targets stationed in close proximity to buildings can easily be missed using conventional analysis methods. By detecting buildings and then estimating their geometry and SAR appearance models, changes in appearance from stored models can be exploited to detect the presence of an inserted target. There is significant skepticism among some in the ATR community about approaching practical target detection in any other way.

Unlike EO imagery, the full geometric structure of a building is not present in SAR imagery. The three prominent features of a building's SAR signature are: (1) bright line(s), resulting from the wall-ground dihedral(s) facing the radar; (2) shadow regions where there is no energy return; and (3) backscatter from small structures on the roof or wall of a building, and/or weak backscatter from rough surfaces.

The algorithm we developed to exploit some of these features is discussed in detail in Sec. 2.1.

1.2 Fast Algorithms for Markov Random Fields

Making progress in the second key technology area has a lot of practical value for target-detection matched filters used in both multi-spectral infrared and synthetic-aperture imagery. It is also very important for super-resolution algorithms that improve target detection/classification performance by removing clutter from these images, which are often severely degraded by speckle and environmental/propagation effects. These are both extremely active areas of research for many workers in the ATR community, partly because many new sensors such as the Global Hawk and Dark Star will be coming on line in the next few years. In addition, existing ATR technologies are being challenged to meet more stringent requirements for target detection and correct-classification probabilities, given a specified false alarm rate, in programs and ACTD's such as SAIP, MSTAR, and FOPEN. The latter, a foliage-penetration low-frequency SAR program, is an excellent example of the increased difficulty of detecting and classifying targets obscured by environmental effects. These effects include HF radio interference as well as the stochastic effects of scattering, blurring and layover induced by the foliage. The lowered range resolution, due to the smaller available radar bandwidth at low frequency, also makes the ATR problem for FOPEN more challenging than that of traditional microwave-frequency SARS, most of which operate in the X-band region or above. The fast algorithms discussed in this report for estimating stochastic target models are applicable to target-detection filters for these problems, including multidimensional imagery such as fully-polarimetric or multifrequency SAR ([1], [2], [3]). In addition, they are also applicable to the challenges of resolution-enhancement, and of removing the radio interference by use of adaptive nulling.

Other examples of the programmatic relevance of the fast-algorithms part of our research can be taken from the set of issues important to SAIP (Semi-Automated IMINIT Processing). This is an ACTD whose focus is to relieve the mounting pressure on human analysts who are being tasked to analyze SAR imagery that arrives at least twice the traditional rate, while the available number of analysts is also shrinking by a factor of 2 to 3. More efficient target-detection algorithms can ease the computational load in the current system, whose single biggest contribution comes from the super-resolution algorithms. In addition, many such SAR systems are stressed by the limited bandwidth available in existing communication links. Stochastic models such as the standard auto-regressive models, or the interpolation models discussed in our research, can be used for image compression, a challenge that will be increasingly important in years to come.

Both the interpolation problem that we have studied and the more common problem of linear prediction are widely-used paradigms for random processes that are frequently encountered in engineering. In the areas of speech processing and adaptive filtering, the use of lattice-type filters to estimate the coefficients of these models is very well-known ([4], [5]). If the reader is familiar with the block diagram for a lattice filter in speech, or with the Levinson algorithm, this can be used as a helpful, intuitive mental model for the kind of problem and approach in the research described in this report. The underlying mathematics is different, but the overall structure and results have a similar flavor, and can be applied to the same areas.

Basically, the problem of Markov random field parameter estimation is the same as optimum linear interpolation, i.e. two-sided or noncausal prediction. It is known that the problem of standard (one-sided) linear prediction can be solved by orthogonalizing the Fourier basis of powers z^k on the unit circle; this is equivalent to finding the Cholesky (triangular) decomposition of the inverse covariance matrix. To solve the interpolation problem, we have extended this theory to orthogonalize a different, Chebyshev-like basis. We have shown the size-n outer interpolation problem in one dimension can be solved in order n-squared steps by a three-term recurrence, somewhat similar to split algorithms. The solution to inner (usual) interpolation can be recovered from outer interpolation by another fast recurrence, equivalent to embedding a Toeplitz-plus-Hankel problem in a larger Toeplitz problem. The most important part of our algorithm research in this area has been to extend this idea to multidimensional problems, since our main focus is in image-processing; more detailed summaries of the algorithm steps are provided in §2.2. We also have shown how orthogonalization can be used to derive other least-squares algorithms such as the Trench algorithms for Toeplitz and Hankel

matrix inversion, and have defined the spectral inner product, with generalized Parseval equality. Another achievement of this research was the introduction of the Chebyshev transform, which is reminiscent of the discrete cosine transform, but also weighted by the effect of the spectral density– this is relevant to the new basis for interpolation.

Because space is limited in this document, only a very high-level summary is being given here in the executive summary, with a somewhat greater level of detail in §2.2. Readers who are interested in obtaining the full technical picture are referred to [6], which contains general tutorial material on "a signal-processing view of Markov random fields," which is helpful background for the topics mentioned next in this summary. As a part of developing the theory of estimation for Markov Random Fields, we have proven a new, "augmented" version of the Wiener-Hopf principle to explain the spectral form of the optimum interpolator for these processes. It is known that the infinite inverse-covariance matrix for such a process has the coefficients of the denominator of the spectral density. We have shown that this is still nearly true for a finite inverse-covariance matrix whose size is greater than the order of the process. The matrix is banded, with the spectral density coefficients appearing in most of the rows; the only exceptions are called edge elements, located near the corners of the matrix. We also have introduced a family of nearly-stationary "triangular-AR" random processes, for which maximum likelihood and least squares estimators are the same, with closed-form expressions for covariance inversion.

The practical significance of this work is that coefficients that describe a random process can be estimated in a particularly fast and stable way. As an example, the basic ideas from this research were applied to the problem of segmenting an image, by means of clustering together small image areas that have the same underlying statistical model. The statistical model, of course, is the Markov random field coefficients, which were calculated for a multidimensional, i.e. color, image; more detail can be found in [6]. Up to now, relatively little research has been done on such vector-valued Markov random fields.

References

[1] M. M. Scheffé. The multicovariance matched filter for target detection in images. Technical Report 126, Laboratory for Engineering Man/Machine Studies, Division of Engineering, Brown University, 1994.

[2] M. M. Scheffé, Michael M. Blane, and David B. Cooper. Multicovariance matched filter for target detection and background recognition. In Oliver E. Drummond, editor, *Signal and Data Processing of Small Targets 1994, Proc. SPIE*, volume 2235, Orlando, FL, 5-7 April 1994.

[3] M. M. Scheffé. Optimal multidimensional distinction between target and background. In Wendell R. Watkins and Dieter Clement, editors, *Characterization, Propagation, and Simulation of Sources and Backgrounds IV, Proc. SPIE*, volume 2223, Orlando, FL, 6-7 April 1994.

[4] Benjamin Friedlander. Lattice methods for spectral estimation. *Proceedings of the IEEE*, 70(9):990–1017, 1982.

[5] Simon Haykin. *Adaptive Filter Theory.* Prentice-Hall, 2nd edition, 1991.

[6] M. M. Scheffé. *Interpolation, Orthogonalization and Random Fields.* PhD thesis, Brown University, Division of Engineering, 1996. Available from University Microfilms, 300 N Zeeb Rd, Ann Arbor MI 48103-1500, 1-800-824-0814.

1.3 The 3L Algorithm for Fitting Implicit Polynomial Curves and Surfaces to Data

Of great importance to a wide of variety Image Understanding and Image Analysis problems is the ability to represent two- and three-dimensional data or objects. Implicit polynomial curves and surfaces are two of the

most useful representations available. Their representational power is evidenced by their ability to smooth noisy data and to interpolate through sparse or missing data. Furthermore, their associated Euclidean and affine invariants are powerful discriminators, making implicit polynomials a computationally attractive technology for recognizing objects in arbitrary positions with respect to cameras or range sensors. We have developed a completely new approach [1, 2] to fitting implicit polynomials to data. The algorithm represents a significant advancement of implicit polynomial technology for three important reasons. First, it is orders of magnitude faster than existing methods. Second, it has significantly better repeatability and numerical stability than current methods. Third, it can easily fit polynomials of high, such as 14^{th} to 18^{th}, degree. This approach provides a completely new way of thinking about and handling implicit polynomials.

References

[1] M. M. Blane, Z. Lei, and D. B. Cooper. The 3L algorithm for fitting implicit polynomial curves and surfaces to data. Technical Report LEMS-160, Division of Engineering, Brown University, June 1996. Also under review for publication in *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[2] Z. Lei, M. M. Blane, and D. B. Cooper. 3L fitting of higher degree implicit polynomials. In *Third IEEE Workshop on Applications of Computer Vision*, Sarasota, Florida, December 1996.

1.4 Personnel Involved in the Research Effort

1. David B. Cooper, Principal Investigator
2. Molly M. Scheffé, Graduate Student
3. Michael M. Blane, Graduate Student
4. Tolga Tasdizen, Graduate Student

1.5 Publications Stemming from the Research Effort

[1] M. M. Scheffé. *Interpolation, Orthogonalization and Random Fields.* PhD thesis, Brown University, Division of Engineering, 1996. Available from University Microfilms, 300 N Zeeb Rd, Ann Arbor MI 48103-1500, 1-800-824-0814.

[2] M. M. Scheffé. Two-sided prediction orthogonalization, and random fields in images. In *IEEE International Conference on Acoustics Speech and Signal Processing*, Atlanta, GA, May 1996. IEEE.

[3] M. Barzohar and D. B. Cooper. Automatic finding of main roads in aerial images by using geometric-stochastic models and estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):707–721, July 1996.

[4] M. M. Blane, Z. Lei, and D. B. Cooper. The 3L algorithm for fitting implicit polynomial curves and surfaces to data. Technical Report LEMS-160, Division of Engineering, Brown University, June 1996. Also under review for publication in *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[5] Z. Lei, M. M. Blane, and D. B. Cooper. 3L fitting of higher degree implicit polynomials. In *Third IEEE Workshop on Applications of Computer Vision*, Sarasota, Florida, December 1996.

1.6 Ph.D. Degrees Granted

Molly M. Scheffé, May 1997

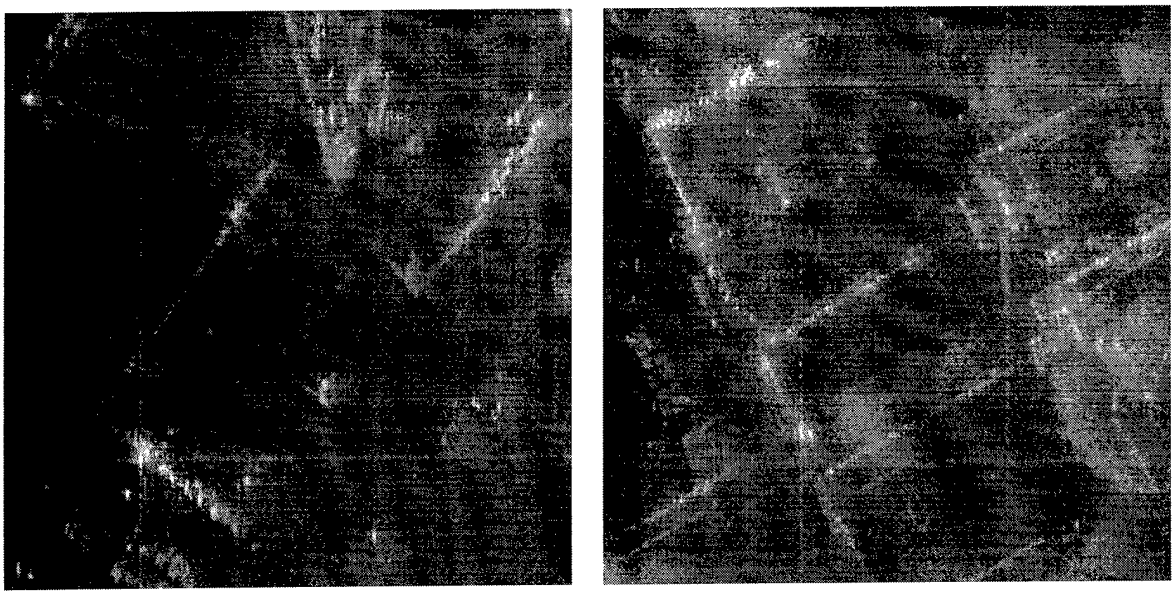


Figure 1: Two detail regions taken from PWF processed SAR image

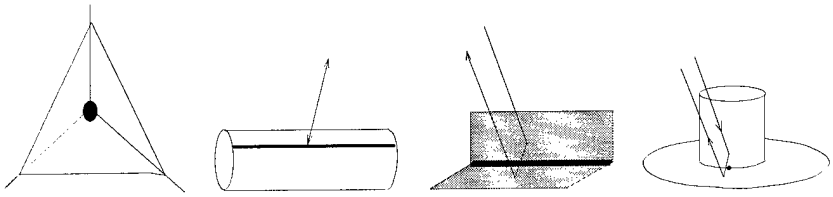


Figure 2: Certain geometric primitives and their SAR signatures

2 Technical Details

2.1 SAR Building Detection in Urban Environments

2.1.1 The Nature of SAR Imagery

The SAR data used in our research consists of various unclassified MIT Lincoln Laboratory ADTS mission passes. Each pass contains four complex, raw-data signals, corresponding to the four polarization channels (HH, HV, VH, and VV) acquired by the sensor. To create an intensity image from these channels, the channels can either be used individually, yielding three separate images, or collectively, yielding a single image. We accomplish the latter using the Polarimetric Whitening Filter (PWF) to produce a minimum-speckle image. The SAR image examples that appear in this report were all PWF processed. We emphasize, however, that the algorithm we developed does not require this; the input is *any* SAR image, regardless of how it was obtained. Figs. 4- 5(a) show a typical (2048 x 512) PWF processed SAR image.

Fig. 1 shows two detail regions taken from Polarimetric Whitening Filter (PWF) processed SAR imagery. The SAR data came from an urban environment, and the detail regions show a mixture of cultural and natural structures. How a building appears in this data depends on its geometry and composition. That is, different scatterers (Fig. 2) on the building give rise to different SAR signatures.

A *dihedral*, such as a wall-ground interface, gives rise to a bright line which appears in the SAR image at a range corresponding to the distance between the radar antenna and the wall-ground interface. A *trihedral*, a corner that is convex to the radar antenna, such as a inset-window corner, gives rise to a small bright

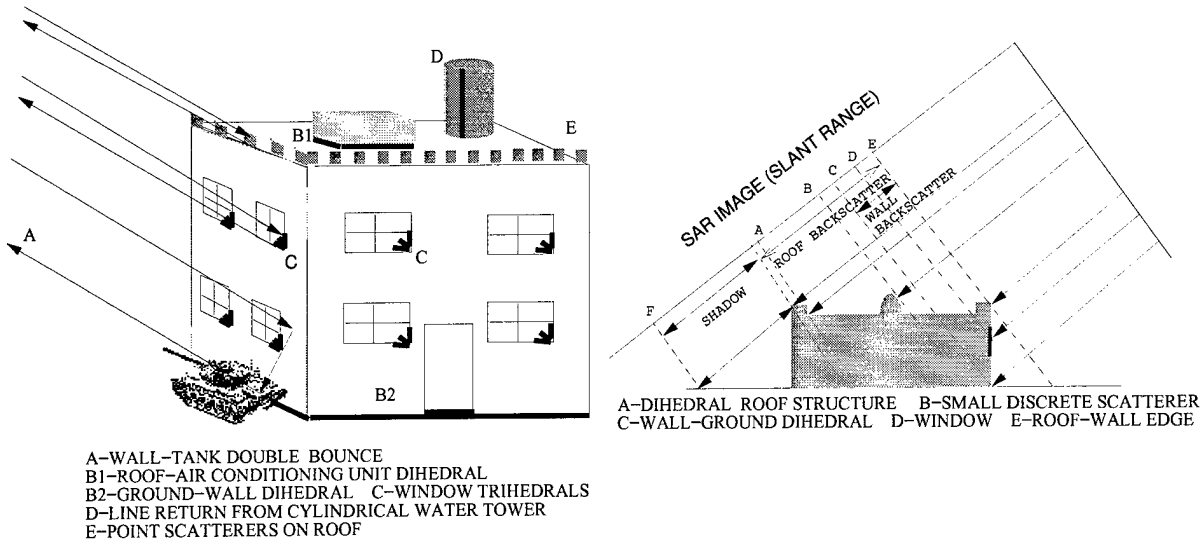


Figure 3: A sketch of some SAR signal returns

spot. A *cylinder*, such as a water tower, can produce a line. Decorative structures on a roof-wall interface are *small discrete scatterers*, which can produce a linear pattern of small spots. Generally there will be backscatter containing speckle (which can often be modeled as a short spatial-correlation Gaussian process) from a surface such as a roof or a rough wall. This backscatter can be bright or it can be very dim. Finally, there are radar shadows, due to occlusion.

Fig. 3 is an example of a building whose geometry and composition would give rise to the hypothetical SAR signature shown, according to the scattering properties described above. It is important to emphasize that the *actual* SAR signature depends strongly on the physical construction of the building. For example, improper joints in window frames might reduce the strength of trihedral returns, or cracks in roofing material might create strong backscatter. Because of this, it was necessary to develop a building detection scheme capable of handling nuances in building signatures.

2.1.2 Discussion of the Algorithm

As mentioned in the Executive Summary, buildings generally have three prominent features in their SAR signatures: (1) a bright line segment corresponding to the dihedral reflection from a wall-ground interface, termed a *crease*; (2) shadow regions on the sides of the building away from the radar; and (3) backscatter caused by windows, or smaller scale objects on the building walls or roof. To detect buildings, we developed a hierarchical approach: we first look everywhere for creases, and then search for supporting clues in the areas surrounding the most likely candidates.

At the heart of the algorithm is a template based crease detector. Here, we use different features to determine if there is a crease located at a particular position and orientation in the SAR image. These features are based on both local and global image information, and are designed to capture a broad range of typical crease characteristics. Most importantly, the algorithm design is modular. Using the output of the crease detector, the reliability of the building detections can be greatly increased by exploiting other aspects of the SAR signature to verify the detections. Under this research program, we developed modules to classify crease geometry, and verify proper shadowing. We were unable to pursue a backscatter module prior to contract end.

A challenge in bright-line-detection in SAR imagery is that the nature of the pixel-level image intensity values can be of various types. A line can appear as a thin ridge, a thin sequence of discrete points, a

concatenation of thin ridges and small blobs, or a wide swath. There are various scattering models that give rise to all of these. The crease finder described in the following paragraphs was designed to be robust to all of these types. In a more sophisticated system, it might be worthwhile to interpret estimated line segments with regard to the type of scattering geometries and surfaces that could have given rise to them.

Creases tend to be composed of locally bright pixels in an image. That is, the pixels along a particular crease vary in intensity about an average value, while pixels off the crease are of lower intensity. As such, we briefly discuss why a simple threshold technique to detect creases is inadequate. In the ideal case, creases would be uniformly bright, and the only brighter pixels would correspond to signatures of trihedral reflectors, strong point reflectors, etc.; off the crease, the "background" intensity would be dramatically lower. However, in the practical case, creases have different average intensities within a particular image. A global threshold would necessarily exclude some creases from detection. Likewise, the difference in intensity between a crease and its background can be highly variable. A local threshold would necessarily introduce false creases. This is particularly evident when natural clutter is present. A crop of trees situated in the correct manner can produce one or more crease-like patches in its signature. The solution lies in combining local information - clues about the geometry and composition of a crease (simple line segment, not too wide, not too broken, and not too misshapen) - with global intensity information to affect a compromise between missed and false creases.

As alluded to above, the global image information we use is a histogram of the intensity values for the entire SAR image. Having computed this, the algorithm processes the (2048 x 512) ADTS derived SAR images by operating on non-overlapping *windows* of size (W x W). W is taken to be a power of two for convenience. In the template based crease detection stage, the templates we use are elongated rectangles of dimension (3 x 64) pixels at different orientations. These values were chosen in accordance with typical crease lengths seen in the data (related to building wall lengths), and are a parameter of the algorithm. Fixing the template length sets W. Here, $W = 64$. The template is applied to the SAR data in a manner as to cover every pixel in every window with at least one template at every orientation. To accomplish this, our orientation resolution is $\frac{\pi}{18}$ radians.

The templates are applied to the underlying SAR image in two ways. First, we use the intensity data to obtain a Constant False Alarm Rate (CFAR) style statistic. That is, we compare the average intensity within the template region to the average intensity in a neighbor around the region. Since a crease is composed of locally bright pixels, we expect a high value for this statistic if there is a crease situated under the template. The statistic is compared to the appropriate false alarm threshold rate, and classified according to outcome. Second, but only if the outcome was class "crease", we create a binarization of the intensity data within the templates. Here, we compute a local histogram of the intensity data within each window and determine its 0.90 quantile. This value is compared to the 0.96 quantile of the global histogram previously computed, and the larger of the two is selected as the binarization threshold. The effect is that the global threshold acts as a lower bound on crease pixel brightness; locally, the threshold is allowed to be greater than this value. These quantiles were selected by extensive empirical study of the ADTS data. They were best able to differentiate creases from other types of returns. The binarization itself consists of comparing the intensity value of each pixel in the template to the threshold. The result is a map which indicates which of the pixels in the template are part of the candidate crease. This allows us to detect creases that do not fill the entire template.

Using this map, we perform a pixel count and do a "gap test" - a structural test of the integrity of the candidate crease. We require that at a minimum, 45 percent of the pixels in the map be "on". Although this appears to be a low number, it is set intentionally to account for thinner creases: the templates are three pixels wide, but creases may be only one or two pixels wide. For the gap test, we search for breaks in the direction of the template, and require that these gaps be no longer than $\frac{1}{4}$ of the template length, or 16 pixels in the typical case. If there were a gap bigger than this size, an explanation is that there are two distinct creases in the template. As a direct consequence of the gap test, it is seen that two creases each smaller than 64 pixels and separated by only 20 pixels would go undetected. To avoid this possibility, we

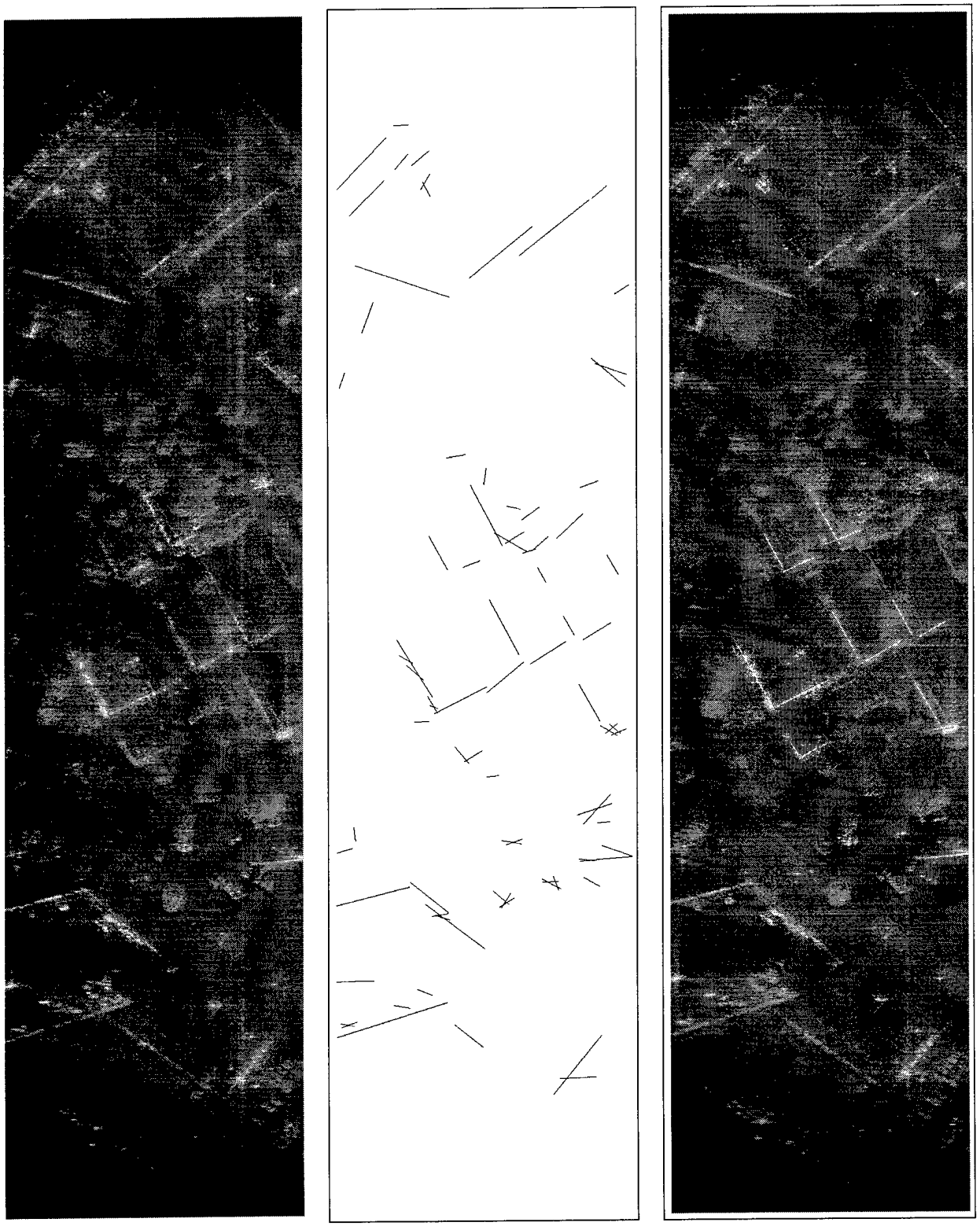
simply repeat the entire process with smaller templates (i.e. 3 x 32). While this increases the amount of computation, superior results are obtained with the gap criteria in place.

If the data in a template passes each of the above tests successfully, it is fully considered a candidate crease. Typically, a true crease will give rise to candidate creases in more than one template. The best statistics will come from the template aligned perfectly with the crease; as templates move away (displacement or orientation) from the crease, their statistics will gradually worsen. Because more than one crease can be present in a window, choosing a single best template is not appropriate. Instead, we select all the best templates which are dissimilar to each other in position and orientation such that they would suggest the presence of multiple distinct creases in the window. We say that two templates are dissimilar if their centers are more than 15 pixels apart or their orientations differ by more than 0.5 radians.

The first pass is complete after all windows have been processed. Note that the windows can be processed in parallel, for a significant computational savings. The second pass is identical to the first, but with smaller templates. This will detect multiple creases too small to pass the first gap test. Before performing the second pass though, any creases detected in the first pass are removed from the SAR image, to prevent their being redetected. Upon completion of the second pass, all the detected creases must be grouped, since creases might easily run for longer than the (3 x 64) pixel template size. Also, this allows us to correct multiple detections of a single crease, by merging creases with very similar positions and orientations.

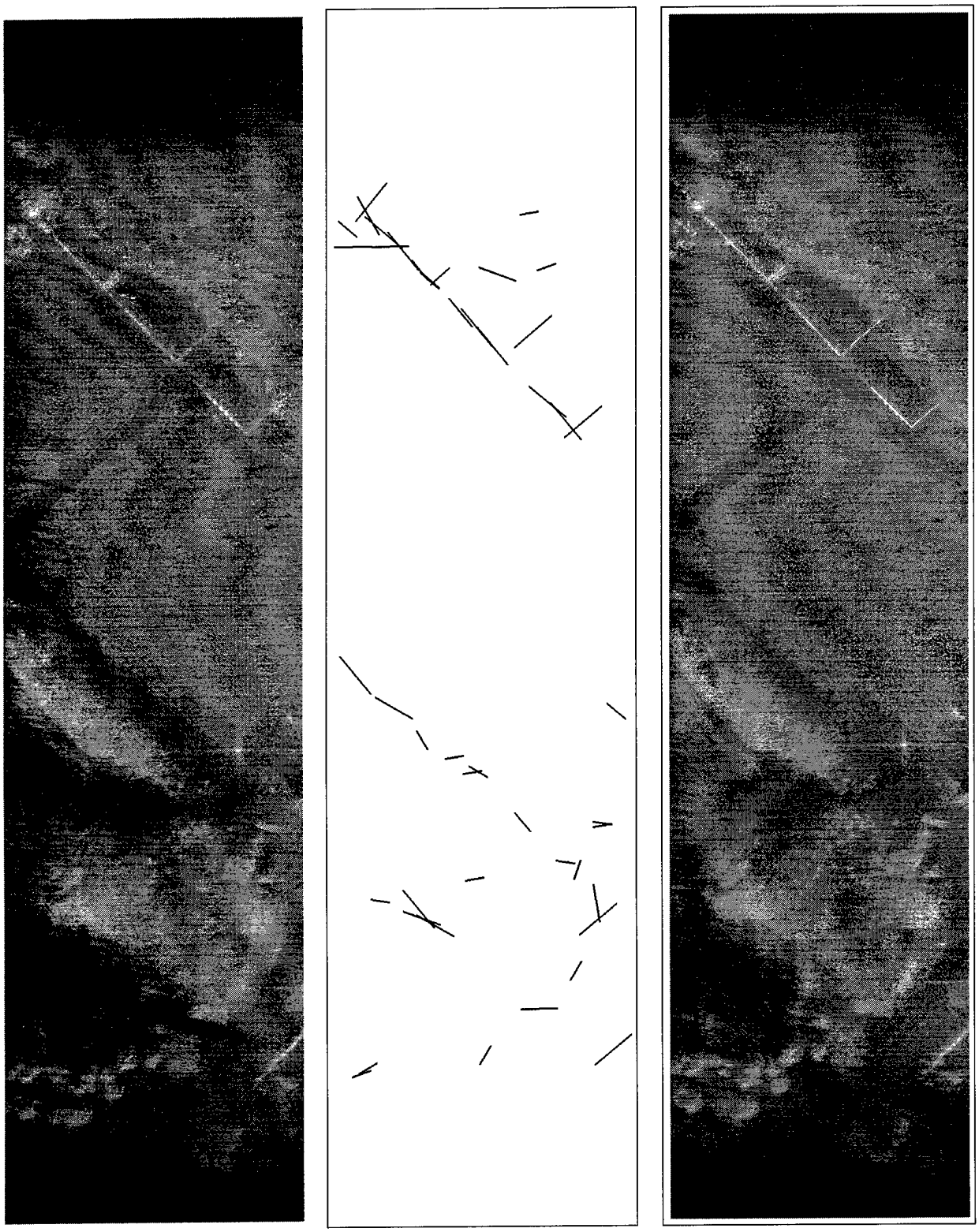
With high likelihood, each of the crease-groups represents a bona fida crease. To best estimate its actual orientation and position, we fit a line to the amalgamated map data for each crease-groups (all the pixels that passed the threshold) using a scatter matrix (i.e. least-square perpendicular distance error). These are the creases shown in Figs. 4- 5(b).

To detect a building, we recognize that a typical rectangular building gives rise to two creases (unless, of course, one of its walls is perpendicular to the line of sight of the radar, which we handle separately). The key is to look for pairs of creases situated at approximate right angles to one another, or, for single creases, at right angles to the line of sight of the radar (apriori known in these images). The final step is to look for shadows behind these creases away from the radar. This is again done by a negated CFAR statistic method since it is expected that shadow regions will be darker than their surroundings. Buildings are detected if all the criteria are met. Figs. 4- 5(c) depict the detected buildings overlayed on the original SAR image.



(a) PWF Processed SAR Image (b) Candidate Creases (c) Detected Buildings

Figure 4: SAR Image of an Urban Area with Detected Buildings



(a) PWF Processed SAR Image (b) Candidate Creases (c) Detected Buildings

Figure 5: SAR Image of an Urban Area with Detected Buildings

2.2 Fast Algorithms for Estimating Markov Random Fields

This section gives a greater level of detail about our work on task two of the research. The material is taken from Scheffe's thesis [1] which is available as a complete reference for those who would like the full technical picture, more background, or additional references to the literature. For this report, we have emphasized intuitive and introductory material, so that at least the reader can see what the basic ideas were and how they relate to other concepts that may be more familiar. To communicate the basic ideas while avoiding some tedious mathematical details, results are often stated just for one-dimensional processes, even though they are valid in two dimensions with a more complicated formulation.

2.2.1 What is a Markov Random Field?

As indicated in the accompanying box, there are at least four important ways of describing spatial random processes known as Markov random fields (MRF). Unfortunately, it is not possible to give a complete tutorial here about these viewpoints and why they are equivalent, but we will at least make a formal definition here sketching the first viewpoint. The next viewpoint, involving least squares estimates of model parameters, is probably most familiar to the general audience. This second viewpoint is described in some detail in §§2.2.3-2.2.6, so nothing further will be said about it here in §2.2.1. Finally, here in §2.2.1, we will give a brief, intuitive introduction to the last two viewpoints. A good deal more information about the last two is available in [1], including an interesting generalization of the Wiener-Hopf theorem. Since our guiding principle here is to present a selection of intuitive and conceptual material (at the possible risk of being too introductory or possibly brief), the Wiener-Hopf material has been omitted.

Summary 2.1 (Four Views of Stationary Gauss-Markov Random Fields)

Listed below are four different mathematical objects that are specified by certain coefficients. Although the mathematical descriptions are different, the coefficients have to be the same for a zero-mean, stationary Gaussian Markov random field, and are referred to as interpolation coefficients or random field parameters.

- coefficients g_i of the data variables y_i in the **Gaussian conditional mean**, $\mu_{x\,|\,\mathbf{y}} = \sum_k g_k\, y_k$; this describes the probability the center variable x takes on a certain value, conditioned on the (finite!) set of relevant neighboring values y_k.
- solution to the problem of finding the **optimum linear interpolation estimator** for the random process : $(\widehat{x}_{\mathbf{g}})_k = \sum_{l\neq 0} g_l\, x_{k+l}$
- coefficients of the inverse **spectral density** $Q(z) = 1/S(z) = q_0\left(1 - \sum_{k\neq 0} g_k z^{-k}\right)$
- entries in the rows of the **inverse covariance** $\mathbf{R}^{-1} = \mathbf{\Sigma}^{-1}$, except for edge effects near the corners of the matrix. This is a **banded** matrix, which is Toeplitz, except for edge effects. The quadratic form $\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{x}$ is an important example of a **potential function**, an equivalent way to specify a Markov random field.

Conditional Mean Approach– Neighbors and Interactions in a Lattice:

Random fields are often described as values x_{ij} in a two-dimensional **lattice**. Intuitively, a **Markov** random field has the property that only *finitely* many neighbors influence the field value at a central location in the lattice. This is illustrated at the left, which shows a central pixel at location (i,j) in a large array of numerical data, that could represent anything from grey-level image values to yields of potato plants in the agricultural field test station. An example of the Markov idea is that the center value $x_{i,j}$ is influenced only by the immediate neighbors to the west, north, south, and east. These neighboring values are labelled y_1, y_2, y_3, y_4 for simplicity in the diagram.

	$x_{i-1,j-1}$	$y_2 = x_{i-1,j}$	$x_{i-1,j+1}$
	$y_1 = x_{i,j-1}$	$\boxed{x = x_{i,j} = ?}$	$y_4 = x_{i,j+1}$
	$x_{i+1,j-1}$	$y_3 = x_{i+1,j}$	$x_{i+1,j+1}$

For this example, the Markov property is expressed by saying that the conditional probability that the center variable $x_{i,j}$ takes on a certain value x depends *only* on the y's, even if a much greater amount of neighboring data is given:

$$pr\left(x_{i,j} = x \,\middle|\, \begin{array}{c}\text{all other pixel}\\ \text{values in image}\end{array}\right) = pr\left(x_{i,j} = x \,\middle|\, \begin{array}{l} x_{i,j-1} = y_1 \\ x_{i-1,j} = y_2 \\ x_{i+1,j} = y_3 \\ x_{i,j+1} = y_4 \end{array}\right) = pr(x \,|\, \mathbf{y}) \tag{2.2.1}$$

This is an example of a *first-order* Markov random field, where the neighbors influencing the center value are no more than the minimum distance away from the center point in the lattice.

A general Markov random field whose order is *at most* n is defined by requiring that for each central lattice site (i,j), only its neighbors of order up to n influence the random field value there:

$$pr\left(x_{i,j} = x \,\middle|\, \begin{array}{c}\text{all other pixel}\\ \text{values in image}\end{array}\right) = pr\left(x_{i,j} = x \,\middle|\, x_{k,l} = y_{k,l} : \begin{array}{l}(k,l) \text{ a } p\text{th-order neighbor}\\ \text{of } (i,j),\ p \le n,\ (k,l) \ne (i,j)\end{array}\right) \tag{2.2.2}$$

For the order of a Markov field to be uniquely defined, we can keep whittling down the set of neighbors on the right until we are beyond the critical size for which the Markov property is true. So in general, an **mth order Markov random field** is defined by taking m to be the minimum of the integers n satisfying the right side of (2.2.2). The values of the neighbors which influence the central pixel value x often will be denoted simply as $\mathbf{y} = (y_1, \ldots, y_k, \ldots)$, using a simple scalar index i, rather than a planar index like (i,j). These will be called **data variables**: given a central site location, the data variables correspond to the neighboring sites which must be included in the conditional probability (2.2.2). So far, the definition we have just made is satisfied by a great variety of random processes which are neither stationary nor Gaussian. However, making these addtional requirements means that the conditional probability in (2.2.2) has to be a linear combination (weighted sum) of the data values y_l. The weights in this sum are the Markov random field coefficients g_l; just as with many other least-squares problems such as prediction, they can be found by solving the appropriate set of normal equations (§2.2.3).

This general definition involving neighbors with finite influence on a lattice applies to an enormous variety of interesting spatial phenomena, ranging from magnetic moments in a lattice, to geographical distributions of human social phenomena, to the distributions of galaxies. For a good deal of such processes, it is not necessary to make the stationary Gaussian assumption mentioned in the box; generalizations of the first and fourth viewpoints still define what a Markov random field is.

Survey– Covariance Inversion, Interpolation and the Spectral Density: Here, we will give an introduction to a very useful relationship among the inverse covariance, the optimal interpolator, and the spectral density of a second-order stationary random process. [1] This relationship does not require that the spectral density denominator be factorizable into "poles" (especially unlikely in 2D), but it does require that it be a symmetric polynomial-type form, which is explained more in [1]. This is a very mild general condition, and is satisfied by most processes studied in signal processing.

As an example of the relationship, we will consider the second-order AR process in one dimension, where the difference equation

$$x_k = ax_{k-1} + bx_{k-2} + w_k, \quad a, b \in \mathbb{R}, \tag{2.2.3}$$

is driven by white noise w_k. The frequency-space version of this difference equation is $(1-az^{-1}-bz^{-2})\,X(z) = W(z)$, where $W(z)$ is the input and $X(z)$ is the output. So the associated transfer function is $H(z) = 1/(1-az^{-1}-bz^{-2})$, and thus the spectral density $S(z)$ for this process must be given by $H(z)\,\overline{H(z)}$:

$$\begin{aligned} S(z) &= \frac{1}{\left(1-az^{-1}-bz^{-2}\right)\left(\overline{1-az^{-1}-bz^{-2}}\right)} \\ &= \frac{1}{(1+a^2+b^2) + a(1+b)\left[z+z^{-1}\right] - b\left[z^2+z^{-2}\right]} \end{aligned} \tag{2.2.4}$$

The denominator here is a good example of one of those symmetric forms in z and $z^{-1} = \overline{x}$ which we just alluded to.

It is interesting that the coefficients in this spectral density occur as the rows of the inverse covariance:

$$\mathbf{R}^{-1} = \begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & 1+a^2+b^2 & a(1+b) & -b & 0 & \cdots \\ \cdots & a(1+b) & 1+a^2+b^2 & a(1+b) & -b & \cdots \\ \cdots & -b & a(1+b) & 1+a^2+b^2 & a(1+b) & \cdots \\ \cdots & 0 & -b & a(1+b) & 1+a^2+b^2 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{2.2.5}$$

This is not an algebraic coincidence, but rather a basic spectral property of Toeplitz matrices, which is discussed in [1].

The reason for including this result in a discussion of Markov random field coefficients should be clear if we look at the optimal interpolator associated with this random process:

$$\widehat{x}_k = \frac{1}{1+a^2+b^2}\left\{a(1+b)\left[x_{k-1}+x_{k+1}\right] - b\left[x_{k-2}+x_{k+2}\right]\right\} \tag{2.2.6}$$

Again, it is not a coincidence that the spectral density coefficients have appeared in the optimal interpolator– this section will show that this actually can be seen as an outcome of a certain kind of Wiener-Hopf theory. All these concepts are so closely related for a Gauss-Markov process that the inverse covariance can be

[1] For this discussion, it is actually not necessary to make the Gaussian assumption, provided the phrase "optimum estimator" is replaced with "optimum linear estimator."

identified with the optimum interpolator, up to a scaling. This is important, because the inverse covariance appears in the argument of the exponential in the basic Gaussian distribution

$$p(\mathbf{x}) = \frac{1}{2\pi^{\frac{n}{2}} |\mathbf{R}|^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{x}} = \frac{1}{Z} e^{V(\mathbf{x})} \tag{2.2.7}$$

This quadratic form $V(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{x}$ is often called a **potential function**, as in the Hammersley-Clifford theorem, a central part of the lore of Markov random fields.

Another example of the relationship between covariance inversion and interpolation comes from the first-order AR process $x_{k+1} = ax_k + w_{k+1}$, where the coefficients in the spectral density

$$S(z) = \frac{1}{(1+a^2) - a[z+z^{-1}]}$$

also occur in the covariance inverse

$$\mathbf{R}^{-1} = \begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & 1+a^2 & -a & 0 & 0 & \cdots \\ \cdots & -a & 1+a^2 & -a & 0 & \cdots \\ \cdots & 0 & -a & 1+a^2 & -a & \cdots \\ \cdots & 0 & 0 & -a & 1+a^2 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Again, the spectral density coefficients also occur in the interpolator for this first-order AR process

$$\widehat{x}_k = \frac{a}{1+a^2}\left\{x_{k-1} + x_{k+1}\right\}$$

The inverse-covariances in the two examples above were both infinite matrices. Some not-very-major corrections have to be made in the upper lefthand and lower righthand corners to apply this idea to finite covariance inversion. We call these **edge effects**, and discuss how to calculate them in [1]. Because of these edge effects, a finite inverse-covariance is prevented from being completely Toeplitz: the elements will not be constant down a diagonal in the vicinity of the top left and bottom right corners. After all, a stationary covariance matrix *has* to be Toeplitz, and generally the inverse of a Toeplitz matrix does not usually have the Toeplitz property. However, a finite Gauss-Markov inverse-covariance is *nearly* Toeplitz: away from the corners, the matrix elements do follow a Toeplitz pattern, with coefficients supplied by the spectral density.

The spectral density (2.2.4) for the second-order AR process turned out to involve symmetric terms in z^{-2}, z^{-1}, ..., z^2. For a general mth order process in one dimension, the denominator will involve terms up to $z^{\pm m}$, which means that rows in the inverse covariance matrix will extend m units to the left and m units to the right of the main diagonal. This says that **for a Gauss-Markov process of order m, the inverse covariance is a banded matrix with $2m+1$ nonzero diagonals. Away from the edges, each row consists of coefficients from the spectral density denominator.** The entries are somewhat different near the northwest and southeast corners, but they do still do not extend more than m steps away from the main diagonal before going to 0. In conclusion, this is an interesting and useful characterization of Gauss-Markov covariances, and it unifies the interpolation problem (= estimation of the random field parameters) with spectral estimation for such processes.

2.2.2 Maximum Likelihood, Least Squares and Realizations

In order to have more space to discuss other topics, just a very brief summary of some contributions in this area will be given here (the details, as usual, are in [1]). First, a theorem about equivalent realizations was proven, which shows the relationship between random processes defined by predictive-type difference equations and ones defined by interpolation-type processes. This theorem offers some insights on the spectral properties of simultaneous auto-regressive processes, which are symmetric difference equations driven by white noise. Another accomplishment was the introduction of a family of nearly-stationary random processes called triangular AR (auto-regressive) processes. These are very close to stationary AR processes, and the parameter-estimation methods of maximum likelihood and least squares are identical for these processes,

which might be surprising. Scheffé showed that there is a simple, closed-form analytical expression for the inversion of a triangular-AR covariance. A discussion in [1] shows that in some sense, that triangular AR covariances are actually "closer" to true stationary AR covariances than the Toeplitz approximation. (The Toeplitz approximation is a slightly-incorrect expression for a Markov random field inverse-covariance, which appears quite widely in the literature). We also gave a new construction of an analytical inverse for a finite banded, triangular Toeplitz matrix, based on "partial fractions for matrices."

2.2.3 Orthogonalization in One Dimension

Introduction and Outline on the Importance of Orthogonality: A good introduction to the basic idea here just consists of looking the familiar covariance matrix in a new way. Basically, all the important properties of the covariance matrix– as used in linear prediction and many other applications– really come from the fact that it is a set of inner products

$$\mathbf{R}^n = \begin{pmatrix} r_0 & r_{-1} & r_{-2} & \cdots & r_{-n+1} \\ r_1 & r_0 & r_{-1} & \cdots & r_{-n+2} \\ r_2 & r_1 & r_0 & \cdots & r_{-n+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{n-1} & r_{n-2} & r_{n-3} & \cdots & r_0 \end{pmatrix} = \begin{pmatrix} \langle z^0, z^0 \rangle & \langle z^0, z^1 \rangle & \langle z^0, z^2 \rangle & \cdots & \langle z^0, z^{n-1} \rangle \\ \langle z^1, z^0 \rangle & \langle z^1, z^1 \rangle & \langle z^1, z^2 \rangle & \cdots & \langle z^1, z^{n-1} \rangle \\ \langle z^2, z^0 \rangle & \langle z^2, z^1 \rangle & \langle z^2, z^2 \rangle & \cdots & \langle z^2, z^{n-1} \rangle \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \langle z^{n-1}, z^0 \rangle & \langle z^{n-1}, z^1 \rangle & \langle z^{n-1}, z^2 \rangle & \cdots & \langle z^{n-1}, z^{n-1} \rangle \end{pmatrix} = \mathbf{G}_z \quad (2.2.1)$$

The covariance matrix $\mathbf{R}^n$ on the left is very familiar as the fundamental ingredient for solving the linear prediction problem, as represented by the normal equations. The matrix on the right may look less familiar, but all the key properties used in signal processing really come from its nature as a collection of inner products. The inner products are essentially just a statement of the Fourier relationship between the covariance elements $(r_k)_k$ and the spectral density $S(z)$. They involve powers of the variable $z = e^{i\theta}$ on the unit circle, which is the natural machinery for expressing a Fourier analysis. So one goal of this section is to make the righthand side matrix more intuitive if it seems unfamiliar, because it is really an effective way of looking at least-squares problems, and is not limited to just prediction problems. Great profit is obtained here from the fact that the ideas and approach can be extended to other sets of functions besides the unit-circle Fourier basis $\{z^k\}_k = \{e^{ik\theta}\}_k$. A matrix $\mathbf{G}$ such as the one on the right, which consists of inner products of functions $G_{kl} = \langle z^k, z^l \rangle$ is called a **Gramian** matrix. More generally, the inner products in a Gramian could be of general basis elements $G_{ij} = \langle \varphi^i(z), \varphi^j(z) \rangle$, which are not necessarily powers of z– in §2.2.5 we will examine another basis, which is ideally suited to solving interpolation problems. Sometimes we denote the basis as a subscript appearing with the Gramian matrix, as in G_z, G_φ, respectively.

The basic message here is that all the important properties of the covariance $\mathbf{R}$ come from the inner product structure on the right, and thus can be generalized to our main interest, the interpolation problem. By contrast, the specialized property of the covariance $\mathbf{R}$ being Toeplitz does *not* generalize to interpolation– this means that the Levinson algorithm cannot just be blindly copied for Markov Random Field problems. What does generalize to interpolation (and other least-squares problems) is the idea that **linear prediction really is an orthogonalization procedure.** This chapter emphasizes the geometric interpretation of this idea, which rests just on properties of inner products, not on more specialized structure of the covariance.

The idea of this orthogonalization procedure (which we call the Szegö Recipe) is that the unit circle basis functions $\{z^k\}_k$ are very nice for bland Fourier series work, but that they fail to be the best *statistical* basis for solving a least squares problem. This is because the original basis functions $\{z^k\}_k$ are not orthogonal with respect to the following basic stochastic (spectral or covariance) inner product, which describes the entries of the two matrices above:

$$\mathrm{E}\,(x_i \overline{x_j}) = r_{i-j} = \langle z^k, z^l \rangle$$

If the righthand side above were δ_{i-j}, then the z^k would have started out already orthogonal, i.e. the covariance matrix $\mathbf{R}$ would be equal to the identity matrix $\mathbf{I}$– this would say the random variables $(x_k)_k$ are uncorrelated (white). But in general, it is quite unlikely to expect that $\mathbf{R} = \mathbf{I}$ for an arbitrary random process; noise is rarely white or uncorrelated. The whole idea of orthogonalization is that a new basis

is created whose Gramian matrix of inner products is the identity **I**– essentially, this is the same idea as **whitening**. Otherwise said, the orthogonalization process can be viewed as a procedure which produces new, statistically independent variables out of the originally dependent (correlated) variables– one could say that these are as well-conditioned as possible, from the numerical point of view.

The chapter of [1] on this material also shows that a quite general orthogonalization problem can be seen as really boiling down to linear prediction. The key generalization is just to replace the unit-circle basis $\{z^k\}_k$ with a more general basis $\{\varphi^k\}_k$, but otherwise all the steps are the same– they really just depend on straightforward inner product manipulations. This generalization is extremely important for solving the interpolation problem. As examples of how useful the orthogonalization viewpoint is, it is used there to give some one-line proofs of important results: the Cholesky decomposition and Szegö's determinant expression for constructing orthogonal polynomials. There are many fascinating relationships between the theory of orthogonal polynomials and problems of linear algebra (see e.g. [2]). The end of this chapter discusses some interconnections between the Gram-Schmidt or QR decomposition, Schur and Levinson-Szegö procedures, and Cholesky decomposition.

Bibliographic Survey: The idea of solving the prediction problem by orthogonalizing the unit-circle powers $\{z^k\}_k$ has appeared here and there in the signal-processing literature– brief treatments can be found in [3], [4]. However, this theory has not been popularized or used frequently; it is likely that the covariance-as-Gramian in (2.2.1) is not familiar to some people. The classic reference on the orthogonal polynomials is Szegö [5]; original work on the unit circle polynomials was also done in the 1930's by Geronimus [6], and the classic with emphasis on probability applications is [7].

The rest of this section introduces a number of concepts which are all very central for the fast algorithm descriptions:

- the geometry of inner products, especially Gramian matrices and the spectral inner product
- what order-recursive equations look like for optimum prediction and optimum interpolation in 1 dimension, as shown in the page-sized box.
- error polynomials, which are the basic tool for describing lattice-type filters/estimation algorithms, and require a set of augmented linear equations
- the notion of generalized prediction, which is an original contribution of this research, and uses Gramian matrices rather than covariance, Toeplitz or Hankel matrices

The Spectral Inner Product: The inner products appearing in the Gramian matrix (2.2.1) are not mysterious– they are just a statement of the familiar Fourier inversion property relating the covariance-/correlation elements r_k to the spectral density $S(z)$:

$$r_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} S(e^{j\theta})e^{jk\theta}\,d\theta = \oint_\Gamma z^k\,S(z)\,\frac{dz}{2\pi jz}$$

Here, Γ is a simple, clockwise-oriented closed contour containing the unit circle. For simplicity, we will be looking at a one-dimensional stationary, zero-mean Gaussian random process $(x_k)_k$. Fourier inversion implies that elements of the covariance matrix can be written as

$$\boxed{\mathbb{E}(x_k\,\overline{x_l}) = R_{kl} = r_{k-l} = \oint_\Gamma z^k\,S(z)\,\overline{z^l}\,\frac{dz}{2\pi jz} = \langle z^k,\,z^l\rangle_S} \qquad (2.2.2)$$

Here, "$\mathbb{E}$" denotes the expectation operator (averaging), and a $\overline{\text{bar}}$ indicates complex conjugation.

Most least-squares problems require computing a mean-square error using the expectation norm in the time domain. For example, the **order-n prediction problem** seeks model coefficients $a_1,\ a_2,\ \ldots, a_n$ to

INTERPOLATION

Data from Past: Data from Future:

$\cdots \quad x_{-3} \quad x_{-2} \quad x_{-1} \quad x_0 \quad x_1 \quad x_2 \quad x_3 \quad \cdots$

$\surd \quad \Uparrow \quad \surd$

$\boxed{?}$

First-Order Interpolation

$$(\widehat{x_0})_g^1 = \overline{g_1^1}\, x_{-1} + g_1^1 x_1$$

$\cdots \quad x_{-3} \quad x_{-2} \quad x_{-1} \quad x_0 \quad x_1 \quad x_2 \quad x_3 \quad \cdots$

$\surd \quad \surd \quad \Uparrow \quad \surd \quad \surd$

$\boxed{?}$

Second-Order Interpolation

$$(\widehat{x_0})_g^2 = \overline{g_2^2}\, x_{-2} + \overline{g_1^2}\, x_{-1} + g_1^2 x_1 + g_2^2 x_2$$

$\cdots \quad x_{-3} \quad x_{-2} \quad x_{-1} \quad x_0 \quad x_1 \quad x_2 \quad x_3 \quad \cdots$

$\surd \quad \surd \quad \surd \quad \Uparrow \quad \surd \quad \surd \quad \surd$

$\boxed{?}$

Third-Order Interpolation

$$(\widehat{x_0})_g^3 = \overline{g_3^3} x_{-3} + \overline{g_2^3} x_{-2} + \overline{g_1^3} x_{-1} + g_1^3 x_1 + g_2^3 x_2 + g_3^3 x_3$$

PREDICTION

Data from Past: Data from Future:

$\cdots \quad x_{-3} \quad x_{-2} \quad x_{-1} \quad x_0 \quad x_1 \quad x_2 \quad x_3 \quad \cdots$

$\surd \quad \Uparrow$

$\boxed{?}$

One-Step Prediction

$$(\widehat{x_0})_a^1 = a_1^1 x_{-1}$$

$\cdots \quad x_{-3} \quad x_{-2} \quad x_{-1} \quad x_0 \quad x_1 \quad x_2 \quad x_3 \quad \cdots$

$\surd \quad \surd \quad \Uparrow$

$\boxed{?}$

Two-Step Prediction

$$(\widehat{x_0})_a^2 = a_2^2 x_{-2} + a_1^2 x_{-1}$$

$\cdots \quad x_{-3} \quad x_{-2} \quad x_{-1} \quad x_0 \quad x_1 \quad x_2 \quad x_3 \quad \cdots$

$\surd \quad \surd \quad \surd \quad \Uparrow$

$\boxed{?}$

Three-Step Prediction

$$(\widehat{x_0})_a^3 = a_3^3 x_{-3} + a_2^3 x_{-2} + a_1^3 x_{-1}$$

minimize the mean-square prediction error

$$(\sigma^2)_a^n \triangleq \mathbb{E}\big(\,|\,x_n - a_1 x_{n-1} - \cdots - a_n x_0\,|^2\big) \tag{2.2.3}$$

A useful property of the spectral inner product is that it can change a time-domain expectation like this into a statement involving just algebra with z's in the frequency domain. For example, in the time domain let $\mathbf{f} = (f_0, f_1, \ldots, f_{n-1})$, $\mathbf{g} = (g_0, g_1, \ldots, g_{n-1})$ be complex coefficient vectors describing the sums of shifted random variables $\sum_{k=0}^{n-1} f_k x_k$, $\sum_{k=0}^{n-1} g_k x_k$. Then the time-domain expectation

$$\mathbb{E}\Big(\sum_{k=0}^{n-1} f_k x_k \sum_{k=0}^{n-1} g_k x_k\Big) = \mathbf{f}^T \mathbf{R} \overline{\mathbf{g}} = \sum_{k,l=0}^{n-1} f_k\, R_{kl}^n\, \overline{g_l} \triangleq \langle \mathbf{f}, \mathbf{g} \rangle_{\mathbf{R}} \tag{2.2.4}$$

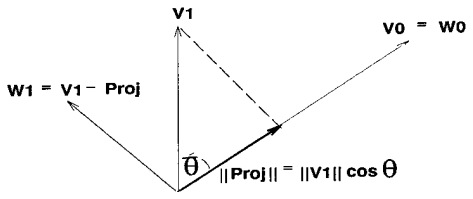
describes an inner product of the two coefficient vectors, since the size-n covariance matrix is positive definite. The **Stochastic Parseval's Relation**

$$\boxed{\langle \mathbf{f},\, \mathbf{g} \rangle_{\mathbf{R}} = \langle F(z),\, G(z) \rangle_{S(z)}} \tag{2.2.5}$$

equates this time domain inner product to a different, frequency-domain inner product of the transforms $F(z) = \sum_{k=0}^{n-1} f_k z^k$, $G(z) = \sum_{k=0}^{n-1} g_k z^k$. (These are polynomial or forward-indexed transforms to avoid confusion with complex conjugates– for interpolation problems, both positive and negative powers of z will be needed). In the frequency domain, each factor $R_{kl} = r_{k-l}$ in the sum from (2.2.4) is replaced with $\langle z^k,\, z^l \rangle_S$ from (2.2.2), verifying (2.2.5):

$$\sum_{k,l=0}^{n-1} f_k\, \overline{g_l}\, \langle z^k, z^l \rangle_S = \oint_\Gamma F(z)\, S(z)\, \overline{G(z)}\, \tfrac{dz}{2\pi j z} = \langle F(z),\, G(z) \rangle_S \tag{2.2.6}$$

The Geometry of Prediction: The Gram-Schmidt process is a familiar way to to orthogonalize a set of vectors, by "subtracting off projections." The very same idea can be used to describe the order-recursive



process of fitting first a one-step predictor, then a two-step predictor and so on, to a set of data. More specifically, the Gram-Schmidt process takes an independent, non-orthogonal set of vectors $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n$, and turns them into a new set of orthogonal vectors $\mathbf{w}_0 = \mathbf{v}_0, \mathbf{w}_1, \ldots, \mathbf{w}_n$ via

$$\mathbf{w}_n = \mathbf{v}_n - \frac{\mathbf{v}_n \bullet \mathbf{w}_{n-1}}{\mathbf{w}_{n-1} \bullet \mathbf{w}_{n-1}}\, \mathbf{w}_{n-1} - \ldots - \frac{\mathbf{v}_n \bullet \mathbf{w}_0}{\mathbf{w}_0 \bullet \mathbf{w}_0}\, \mathbf{w}_0$$

Note that here the (usual, Euclidean) dot product of complex vectors here is $\mathbf{a} \bullet \mathbf{b} = \sum_{k=0}^{n} a_k\, \overline{b_k}$. But both the geometry and the algebra apply to other dot products like $\langle \cdot, \cdot \rangle_{\mathbf{R}}$ or $\langle \cdot, \cdot \rangle_{S(z)}$. The latter is a dot product of functions on the circle rather than vectors.

Instead of starting with a basis of vectors $\{\mathbf{v}_k\}_k$, the process for prediction models starts with the basis of powers $\{z^k\}_k$ on the unit circle [8]. This basis is *not* orthogonal with respect to the spectral inner product $\langle \cdot, \cdot \rangle_S$ unless the covariance $\mathbf{R}$ happens to be the identity matrix (white noise). By orthogonalizing the "vectors" $z^0, z^1, \ldots, z^n$, we are *whitening* the covariance matrix, i.e. finding its innovations or Cholesky decomposition. Instead of being denoted by "$\mathbf{w}_k$," the new basis consists of **error polynomials**

$$e^k(z) \triangleq z^k - a_1 z^{k-1} - \cdots - a_n z^0 \tag{2.2.7}$$

Note that the new, orthogonalized version of z^k is expanded out in terms of the *original* basis $\{z^k\}_k$, whereas the Gram- Schmidt process expands out a new $\mathbf{w}_k$ in terms of the *new*, orthogonalized basis of lower-order $\mathbf{w}$'s. Also, the Gram-Schmidt process is less efficient than modern ways to compute error polynomials because it requires k dot products at step k, whereas say the Levinson algorithm only requires 2. But the Levinson

process and the Gram-Schmidt process *are* equivalent mathematically– this can be verified by substituting an expansion for the new basis in terms of the old.

Prediction Compared with Interpolation: In a general estimation problem, one might be trying to forecast a certain value of a random process using a linear sum of other known values. For example, nth-order linear prediction looks to the n most recent past values to construct an estimate:

$$(\widehat{x}_a)_k^n = a_1^n x_{k-1} + a_2^n x_{k-2} + \ldots + a_n^n x_{k-n} \tag{2.2.8}$$

Here the $\widehat{\text{hat}}$ denotes that an estimate is being made, and k is a general (discrete) time index. Since we are assuming stationarity, it is often convenient to take $k = 0$. By contrast, second-order interpolation, for example, involves two neighbors from the past, and two from the future:

$$(\widehat{x_0})_g^2 = g_{-2}x_{-2} + g_{-1}x_{-1} + g_1 x_1 + g_2 x_2 \tag{2.2.9}$$

The page-sized figure illustrates a number of examples of prediction and interpolation of different orders. Note that the tradition for defining the model order for interpolation uses the number of neighbors on a side, not the total number of neighbors. The letter "a" will be reserved for prediction coefficients, and "g" for prediction coefficients ("β" is also common for Markov fields). The kind of model is indicated as a subscript, and the model order as a superscript.

The Normal Equations in Frequency Space: The normal equations describe how to find the optimum, minimum-error estimator coefficients in terms of the covariance elements r_k. They can be expressed in terms of the **error process**, which for nth order prediction is defined as

$$\left(e_a^n\right)_k \triangleq x_k - (\widehat{x}_a)_k^n = x_k - a_1^n x_{k-1} - \ldots - a_n^n x_{k-n} \tag{2.2.10}$$

The coefficients of this error process will also be denoted as the **error vector** $\mathbf{e}_a^n = (1, -a_1^n, \ldots, -a_n^n)$; these are the same as the coefficients of the error polynomial (2.2.7). We do not follow the tradition of incorporating the minus sign into the coefficients, because we will want to define non-predictive kinds of errors.

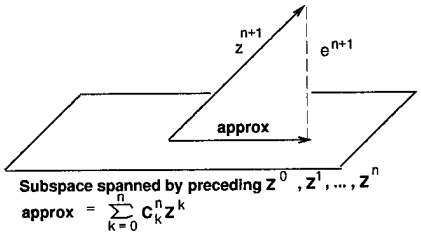
The normal equations (see [9], §13.1) say that this error process must be uncorrelated with each of the variables x_k used to make up the estimate. For prediction, this can be written:

$$\begin{aligned} 0 &= \mathbb{E}\Big(\left[x_n - a_1 x_{n-1} - \cdots - a_n x_0 \right] x_k \Big) \\ &= \mathbb{E}(e_a^n \, \overline{x_k}) = \mathbb{E}(x_k \, \overline{e_a^n}) \quad k = 1, 2, \ldots, n \end{aligned} \tag{2.2.11}$$

Expanding out the expectations with $\mathbb{E}(x_k \overline{x_l}) = r_{k-l}$ would give the familiar Yule-Walker equations. Our interest, though is to express these relations in frequency space, so we replace $\mathbb{E}(x_k \overline{x_l})$ with $\langle z^k, z^l \rangle_S$. What appears is the error polynomial (2.2.7) for prediction:

$$0 = \langle z^n - a_1 z^{n-1} - \cdots - a_n z^0, z^k \rangle_S = \langle e_a^n(z), z^k \rangle \tag{2.2.12}$$

So the frequency-space normal equations *ask that the error polynomial $e_a^n(z)$ be orthogonal to all the preceding basis elements* $z^0, z^1, \ldots, z^{n-1}$. This is why we can identify prediction with the Gram-Schmidt process to orthogonalize the basis elements $\{z^k\}_k$. It also explains why the covariance is the same as the Gramian of inner products in (2.2.1).



Theorem 2.2 (Szegö Recipe) *The $e^n(z)$ error polynomials can be obtained by orthogonalizing the basis $\{z^0, z^1, z^2, \ldots, z^n\}$ with respect to the spectral inner product $\langle \cdot, \cdot \rangle_S$.*

Although usually the indices for prediction go backwards in time, it is much more convenient to have the indices agree with the coefficient of z. We will change notation from "a" to "c," writing the error polynomial as

$$\boxed{e_c^n = z^n - \sum_{k=0}^{n-1} c_k^{n-1} z^k, \quad c_k^n \triangleq a_{n-k}} \tag{2.2.13}$$

These ideas about orthogonalizing error polynomials are not limited to the basis $\{z^k\}_k$, nor to the spectral inner product $\langle z^k, z^l \rangle_S$. A version of these "Yule-Walker" equations occurs in every book on how to orthogonalize a polynomial basis (e.g. [10] §1.3), but usually without making the connection with linear prediction. A matrix system expressing the frequency space normal equations for orthogonalization is

$$\begin{pmatrix} \langle z^0, z^0 \rangle & \langle z^1, z^0 \rangle & \cdots & \langle z^{n-1}, z^0 \rangle \\ \langle z^0, z^1 \rangle & \langle z^1, z^1 \rangle & \cdots & \langle z^{n-1}, z^1 \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle z^0, z^{n-1} \rangle & \langle z^1, z^{n-1} \rangle & \cdots & \langle z^{n-1}, z^{n-1} \rangle \end{pmatrix} \begin{pmatrix} a_n^n \\ a_{n-1}^n \\ \vdots \\ a_1^n \end{pmatrix} = \begin{pmatrix} \langle z^n, z^0 \rangle \\ \langle z^n, z^1 \rangle \\ \vdots \\ \langle z^n, z^{n-1} \rangle \end{pmatrix}$$

A linear system like this will be called the **generalized prediction equations**. It applies to very general orthogonalization problems– the basis need not be the $\{z^k\}_k$, and the inner products do not have to be spectral inner products. Note that the a indices run "backwards" here, but the c indices would go forwards. For brevity, the Gramian matrix of inner products above will also be denoted by $\boxed{G_{kl} = \langle z^k, z^l \rangle}$. The minimum mean-square error associated with the normal equations

$$\begin{aligned} (\sigma^2)_c^n &= \mathbb{E}(e_c^n \, \overline{x_n}) \\ &= \langle z^n, z^n \rangle - c_1^{n-1} \langle z^{n-1}, z^n \rangle \cdots - c_{n-1}^{n-1} \langle z^0, z^n \rangle \end{aligned} \tag{2.2.14}$$

can be incorporated with the above into a larger system, the **augmented** generalized prediction equations:

$$\underbrace{\left(\begin{array}{ccc|c} G_{0,0} & \cdots & G_{n-1,0} & G_{n,0} \\ G_{0,1} & \cdots & G_{n-1,1} & G_{n,1} \\ \vdots & \ddots & \vdots & \vdots \\ G_{0,n-1} & \cdots & G_{n-1,n-1} & G_{n,n-1} \\ \hline G_{0,n} & \cdots & G_{n-1,n} & G_{n,n} \end{array}\right)}_{(\mathbf{G}_z^n)^T} \underbrace{\left(\begin{array}{c} -c_0^{n-1} \\ -c_1^{n-1} \\ \vdots \\ -c_{n-1}^{n-1} \\ \hline 1 \end{array}\right)}_{\mathbf{e}_c^n} = \left(\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ \hline (\sigma^2)_c^n \end{array}\right) \tag{2.2.15}$$

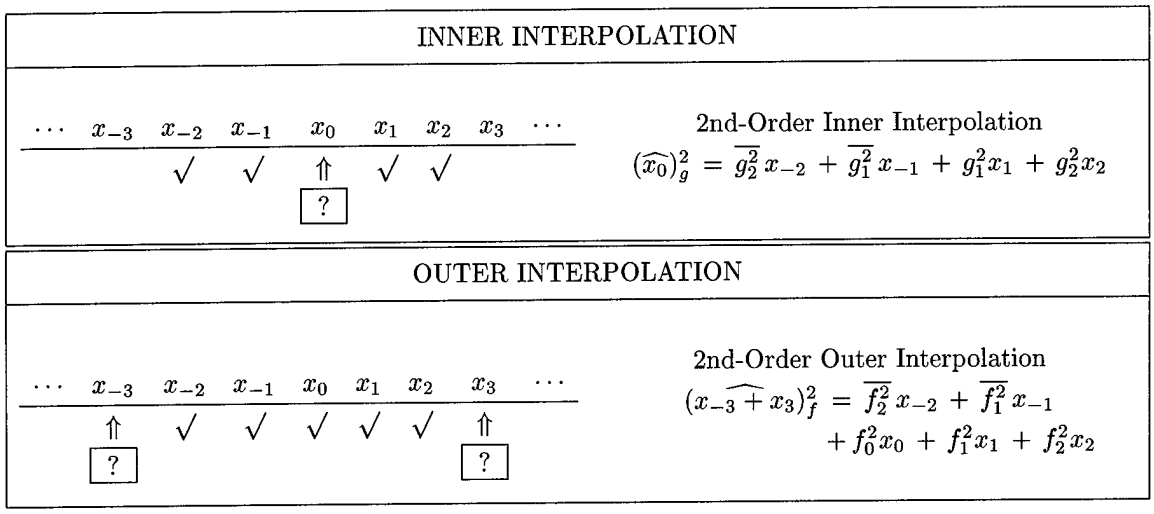
where the error polynomial coefficients appear in the error vector $\mathbf{e}_c^n$ on the left-hand side.

2.2.4 Interpolation in 1D: a Levinson-style Approach

General Preview– Estimating an Optimal Interpolator: In this research, we considered two different approaches to the interpolation problem. Both of these approaches the unifying theme of fitting a non-causal, two-sided model to data for the random process $(x_k)_k$; the page-sized figure in §2.2.3 gives a graphical comparison between two-sided (interpolation) models and one-sided (predictive) models, which are more common in mainstream signal processing.

First, in §2.2.4, we will look at a **Levinson-style approach** to the interpolation problem. This is very similar to the usual machinery in mainstream signal processing for predictive-type problems, which makes it particularly straightforward. Basically, this approach changes a size-n interpolation problem into a size-$(2n+1)$ standard prediction problem. This theme has also appeared in [11], [12], and other work mentioned in the survey below. An original contribution of this research is the basic strategy is of considering *two* kinds of interpolation problems, which are shown in the box.

The usual meaning that people attach to the word "interpolation" corresponds to the first model, called **inner interpolation**: we try to guess the value of a random process at a central point, given data from



both its immediate neighbors to the left and neighbors to the right. The weighting coefficients assigned to the neighbors in this inner interpolation model are denoted g_i^n. The number of neighbors determines the order of the model– it is easy to show that the data must be used symmetrically. **Outer interpolation**, on the other hand, starts from a complete interval of data where the center point x_0 is *not* missing, and tries to guess a value for the average value lying just outside the data interval. Outer interpolation coefficients will be denoted f_i^n.

The interpolation problem can then be solved using the following basic principles:

Three Interpolation Principles

- *Outer interpolation is a special case of generalized (forward) prediction.*
- *Inner interpolation is a case of generalized postdiction (= backward prediction).*
- *A generalized postdiction problem can be solved using the solution to generalized prediction, but not vice versa. Thus, the standard = inner interpolation problem can be solved in terms of the solution to outer interpolation.*

A drawback associated with this first, Levinson-style approach is that it is inefficient: it replaces a size-n problem with a size-$(2n+1)$ problem, requiring twice as much storage and computation as other algorithms. So next, in §2.2.5, a second, more efficient and economical approach is introduced *for real-valued interpolation* problems. This approach is best referred to as an orthogonal-polynomial type approach.

Survey of other work in this area: In one-dimensional signal processing, it has long been known that interpolation problems, which involve a non-causal, symmetric model, require the use of Toeplitz-plus-Hankel matrices (see [13], [12], and the survey in [14]). This T+H problem is particularly important in applications where linear phase is required. Some of the early work in this area ([13], [12]) went the route of mapping a T+H matrix onto a size-$(2n+1)$ Toeplitz matrix; this is also essentially the concept presented in [11], which first presented a lattice filter concept for interpolation. Many fast, order $\mathcal{O}(n^2)$ algorithms have been formulated for the solution of these one-dimensional Toeplitz-plus-Hankel problems (in addition to the above, see also [11], [15], [16], [17], [18], [19], [20], [21].) In principle, block versions of these Toeplitz-plus-Hankel algorithms could easily be applied to solve standard Markov Random Field estimation problems for images. Yet the use of such fast interpolation algorithms seems to be unheard-of in image processing, which

is surprising. The state of the art in image processing is very different than the situation for one-dimensional signal processing, where the use of fast algorithms for linear prediction (causal, one-sided models) is almost universal, and covers an enormous variety of applications.

We conclude this section with a statement of our basic result. Since the Levinson algorithm is so well-known, we do not give the detailed description of the steps to find the $\mathbf{a}^m$ and $\mathbf{b}^m$. These coefficients respectively are the solutions to the forward prediction problem (forecasting) and the backward prediction problem (postdiction = backcasting).

Propostion 2.3 (Outer Interpolation: Levinson Solution) *The solution* $\mathbf{f}^n$ *to the outer interpolation problem can be obtained from the size-*$(2n+1)$ *solutions* $\mathbf{a}^{2n+1}, \mathbf{b}^{2n+1}$ *to the prediction and postdiction problems:* [2]

$$\mathbf{f}^n = \mathbf{J}\mathbf{a}^{2n+1} + \mathbf{b}^{2n+1} \tag{2.2.1}$$

Algorithm 2.4 (Recursion for Inner Interpolation) *Let* $(x_k)_k$ *be a* **real** *stochastic process with all covariance blocks* $\mathbf{R}^n$ *invertible, and let* $\mathbf{a}^m$ *denote the solution to the forward prediction problem of size* m *(obtained e.g. from the Levinson recursion). Then the solution to the size-*n *least-squares* **inner interpolation problem**

$$(\widehat{x_0})_g^n = \sum_{i=1}^{n} g_i^n \left(x_i + x_{-i}\right) \tag{2.2.2}$$

can be calculated from inductively as:

Step One: Get new $(\sigma^2), \Delta$, *using coefficients from preceding stage. By symmetry, only half the entries of these dot products need be computed:*

$$\begin{aligned} (\sigma^2)_f^{n+1} &= 2(r_0 + r_{2n+2}) - (\mathbf{r}^{2n+1} + \mathbf{J}\mathbf{r}^{2n+1})^{\mathrm{T}} (\mathbf{a}^{2n+1} + \mathbf{J}\mathbf{a}^{2n+1}) \\ \Delta_g^{n+1} &= (\mathbf{r}^{2n+1} + \mathbf{J}\mathbf{r}^{2n+1})^{\mathrm{T}} \mathbf{e}_g^n \end{aligned} \tag{2.2.3}$$

Step Two: find new d **and** c**:**

$$d^{n+1} = -\frac{\Delta_g^{n+1}}{(\sigma^2)_f^{n+1} + 2a_{n+1}^{2n+1}\Delta_g^{n+1}} \tag{2.2.4}$$

$$c^{n+1} = 1 + 2d^{n+1}a_{n+1}^{2n+1} \tag{2.2.5}$$

Step Three: Obtain new, size-$(n+1)$ **Interpolation Coefficients**:

$$\begin{pmatrix} g_1^{n+1} \\ g_2^{n+1} \\ \vdots \\ g_n^{n+1} \\ g_{n+1}^{n+1} \end{pmatrix} = -d^{n+1} \begin{pmatrix} a_{n+2}^{2n+1} + a_n^{2n+1} \\ a_{n+3}^{2n+1} + a_{n-1}^{2n+1} \\ \vdots \\ a_{2n+1}^{2n+1} + a_1^{2n+1} \\ 1 \end{pmatrix} + c^{n+1} \begin{pmatrix} g_1^n \\ g_2^n \\ \vdots \\ g_n^n \\ 0 \end{pmatrix} \qquad \square \tag{2.2.6}$$

To introduce the two-dimensional version of this algorithm in §2.2.6, the expression (2.2.6) in step three will also be written in terms of error vectors, that is, the coefficients of error polynomials. This relationship, which is at the heart of the recursion, can also be expressed more briefly as

$$\boxed{\mathbf{e}_g^{n+1} = d^{n+1}\,\mathbf{e}_f^{n+1} + c^{n+1}\,\underset{\longleftrightarrow}{\mathbf{e}_g^n}} \tag{2.2.7}$$

Here the coefficients in the error vectors are given explicitly by:

$$\mathbf{e}_g^n = \left(-g_{-n}^n, -g_{-n+1}^n, \ldots, -g_{-1}^n, 1, -g_1^n, -g_2^n, \ldots, -g_n^n\right) \tag{2.2.8}$$

$$\underset{\longleftrightarrow}{\mathbf{e}_g^n} = \left(0, -g_{-n}^n, -g_{-n+1}^n, \ldots, -g_{-1}^n, 1, -g_1^n, -g_2^n, \ldots, -g_n^n, 0\right) \tag{2.2.9}$$

$$\mathbf{e}_f^{n+1} = \left(1, -\overline{f_n^n}, -\overline{f_{n-1}^n}, \ldots, -\overline{f_1^n}, -f_0^n, -f_1^n, -f_2^n, \ldots, -f_n^n, 1\right) \tag{2.2.10}$$

[2] Here $\mathbf{J}$ is the canonical flip matrix, with 1's down the anti-diagonal.

2.2.5 1D Interpolation by Orthogonal Polynomials: Real Case

Preview– Fast Algorithms Derived From Orthogonal Function Theory: In §2.2.3, we saw that the most basic features of the linear prediction problem did not actually depend specifically on having a Toeplitz covariance matrix. Instead, what really counts is that the matrix in the prediction system is a *Gramian* matrix, made up out of spectral inner products of basis functions $\langle z^i, z^j \rangle_S$. This message applies equally well to estimating an optimal interpolator, not just to the more familiar problem of estimating an optimal predictor. The basic linear system to solve for interpolation does not involve a covariance matrix, but rather a Toeplitz-plus-Hankel matrix. The two linear systems, for prediction and for outer interpolation, do not look totally alike, because they correspond to *different* least-squares problems. Yet from the Gramian viewpoint the prediction system and the interpolation system are essentially identical. The only change is that the Fourier basis $\{z^k\}_k$ of unit circle powers has been replaced with a new real-interpolation basis $\{\varphi^k\}_k = \left\{\frac{1}{2}(z^k + z^{-k})\right\}_k$, which are basically cosine functions rather than powers of z. Note that these are not actually polynomials in z (although they are laurnomials in z and z^{-1}), so the classical theory of orthogonal polynomials has to be generalized somewhat to work here. But otherwise, the mathematics is fairly similar to the standard prediction problem, where z's have been replaced by φ's. This shows the value of the concept of generalized prediction: both the systems in the righthand column are cases of a generalized prediction problem, and they differ only in which basis is being considered. They can be solved by the same general principles; the Szegö recipe applies equally well to both.

The more detailed discussion in this chapter of [1] begins with an examination of the equivalence among the original size-$2n+1$ outer interpolation system in the previous section, the more economical Toeplitz-plus-Hankel version of this system, and the Gramian view of these as generalized prediction. It is also shown how inner interpolation can be viewed as a generalized *post*diction problem, and a very general way of solving postdiction problems in terms of prediction problems. Next there is a further exploration of the basis $\{\varphi^k\}_k = \left\{\frac{1}{2}(z^k + z^{-k})\right\}_k$, since it is only a change of variables away from the Chebyshev polynomials. Many familiar Fourier facts can immediately be translated into Chebyshev properties by this change of variables, such as the generalized Parseval's equality. We also show that with this change of variables, usual transforms– like the polynomial, discrete/continuous Fourier or z transforms– turn into a new kind of transform which we call the Chebyshev transform. To save space in this report, we have omitted both the Chebyshev transform and also an interesting illustration where the theory can be applied in a "turn-the-crank" way to obtain the Trench algorithm for Hankel matrix inversion in just a few lines, rather than in the dozen pages that Trench required (see [22]). This example shows that our theory really is more general than standard linear prediction, and can be used to derive important practical results. In this report, we do indicate how interpolation can be mapped into generalized prediction, and give a fast, three-term recurrence for generalized prediction problems. This recurrence algorithm is reminiscent of the Chebyshev second-order recurrence, but actually does not specifically require that the basis functions $\{\varphi^k\}_k$ be polynomials. It will work for *any* problem defined with a Gramian inner-product structure, for example the Hankel inversion problem.

Various other work on the Toeplitz-plus-Hankel problem was cited in §2.2.4. There does not appear to be any published work solving the interpolation problem by orthogonalizing a basis of functions the way that is done here, even though it is well-known how to solve the prediction problem by orthogonalizing the unit-circle basis.

A Basis for the Interpolation Problem: A consequence of stationarity is the center conjugate symmetry property of the interpolation coefficients: $\boxed{g_k = \overline{g_{-k}}}$. From now on, we will assume the process $(x_k)_k$ is *real-valued*, so that the terms can be grouped together in the normal equations for interpolation. For example,

the mean-square error expression involves

$$\mathbb{E}_k\{|x_k - \sum_{0<|i|\le n} g_i^n x_{k+i}|^2\} = \|z^0 - \sum_{l=1}^{n} g_l^n (z^l + z^{-l})\|^2_{S(z)}$$
$$= \|\tfrac{1}{2}(z^0 + z^{-0}) - \sum_{l=1}^{n} (2\,g_l^n)\cdot \tfrac{1}{2}(z^l + z^{-l})\|^2_{S(z)}$$

This suggests that rather than using the unit-circle basis $\{z^k\}_k$, the interpolation problem can best be expressed in terms of a new symmetrical basis $\{\varphi^k\}_k$, defined as

$$\boxed{\varphi^k(z) = \tfrac{1}{2}\{z^k + z^{-k}\} = \tfrac{1}{2}\{e^{jk\theta} + e^{-jk\theta}\} = \cos(k\theta)} \tag{2.2.1}$$

The factor of $\frac{1}{2}$ was introduced to make the new basis look more like other familiar expressions. With the substitution $u = \cos(\theta)$, these basis functions in fact do become the Chebyshev polynomials $T^k(u)$. The factor of $\frac{1}{2}$ can be counterbalanced by defining $\boxed{d^k = 2g_k}$. Basically, all the constructions we made for prediction using the circle basis $\{z^k\}_k$ now can be carried over to the cosine basis $\{\varphi^k\}_k$ for interpolation. For prediction, the error vectors $\mathbf{e}_a^n$, for example, involved the very latest z^n as a sum of $z^0, z^1, \ldots, z^{n-1}$. Now for interpolation, the spectral normal equations ask for the constant basis vector $\varphi^0(z)$ to be expressed as a growing sum of $\varphi^1, \varphi^2, \ldots, \varphi^n$:

$$\begin{aligned} 0 &= \langle e_d^n(z), \varphi^k\rangle, \quad 1 \le k \le n \\ &= \langle \varphi^0, \varphi^k\rangle - d_1\langle \varphi^1, \varphi^k\rangle - \ldots - d_n\langle \varphi^n, \varphi^k\rangle \\ (\sigma^2)_d^n &= \langle e_d^n(z), \varphi^0\rangle \\ &= \langle \varphi^0, \varphi^0\rangle - d_1\langle \varphi^1, \varphi^0\rangle - \ldots - d_n\langle \varphi^n, \varphi^0\rangle \end{aligned} \tag{2.2.2}$$

The corresponding augmented matrix system looks like backward prediction:

$$\underbrace{\left(\begin{array}{c|ccc} \langle\varphi^0,\varphi^0\rangle & \langle\varphi^1,\varphi^0\rangle & \cdots & \langle\varphi^n,\varphi^0\rangle \\ \hline \langle\varphi^0,\varphi^1\rangle & \langle\varphi^1,\varphi^1\rangle & \cdots & \langle\varphi^n,\varphi^1\rangle \\ \langle\varphi^0,\varphi^2\rangle & \langle\varphi^1,\varphi^2\rangle & \cdots & \langle\varphi^n,\varphi^2\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle\varphi^0,\varphi^n\rangle & \langle\varphi^1,\varphi^n\rangle & \cdots & \langle\varphi^n,\varphi^n\rangle \end{array}\right)}_{\left(\mathbf{G}_\varphi^{[0,n]}\right)^T} \underbrace{\left(\begin{array}{c} 1 \\ \hline -d_1^n \\ -d_2^n \\ \vdots \\ -d_n^n \end{array}\right)}_{\mathbf{e}_d^n} = \left(\begin{array}{c} (\sigma^2)_d^n \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{array}\right) \tag{2.2.3}$$

For prediction, the Gramian matrix $\mathbf{G}_z$ for the basis $\{z^k\}_k$ turned out to be the (Toeplitz) covariance matrix $\mathbf{R}$ in (2.2.1). By contrast, the Gramian matrix $\mathbf{G}_\varphi$ for interpolation is a Toeplitz-plus-Hankel (T+H) matrix:

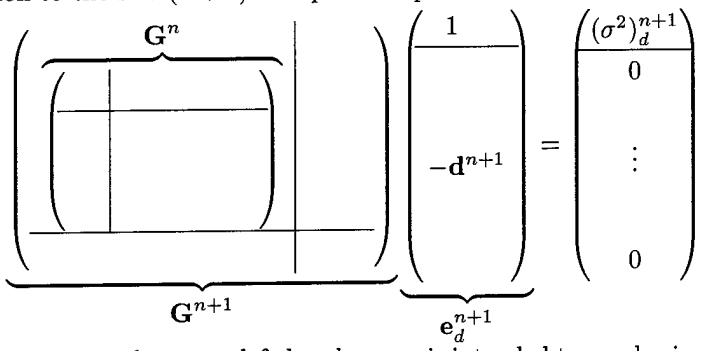
$$\boxed{\langle\varphi^k,\varphi^l\rangle_{S(z)} = \tfrac{1}{2}\Big(r_{k-l} + r_{k+l}\Big) = \tfrac{1}{2}\Big(R_{kl} + H_{kl}\Big)} \tag{2.2.4}$$

where the Hankel matrix is defined by $H_{kl} = r_{k+l}$. So an equivalent way of expressing the augmented normal equations (2.2.3) for interpolation is

$$\left(\begin{array}{c|ccc} r_0 + r_0 & r_1 + r_1 & \cdots & r_n + r_n \\ \hline r_1 + r_1 & r_0 + r_2 & \cdots & r_{n-1} + r_{n+1} \\ r_2 + r_2 & r_1 + r_3 & \cdots & r_{n-2} + r_{n+2} \\ \vdots & \vdots & \ddots & \vdots \\ r_n + r_n & r_{n-1} + r_{n+1} & \cdots & r_0 + r_{2n} \end{array}\right) \left(\begin{array}{c} \frac{1}{2} \\ \hline -g_1^n \\ -g_2^n \\ \vdots \\ -g_n^n \end{array}\right) = \left(\begin{array}{c} \frac{1}{2}(\sigma^2)_g^n \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{array}\right)$$

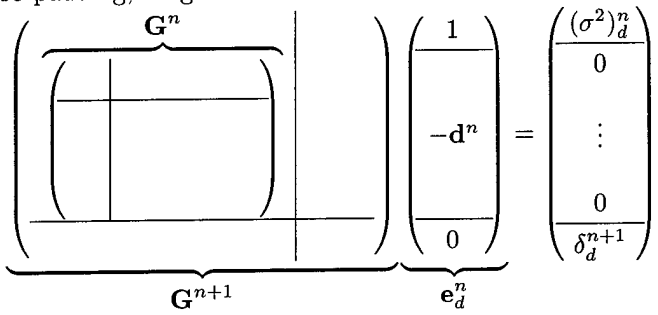
A Recursion for Interpolation: We can find the best least-squares interpolation coefficients $\mathbf{g}$ by "fitting a smaller covariance/Gramian into a larger one," in a way similar to [11]; unlike [11], though, we do not

map the problem to the size-$(2n+1)$ Levinson algorithm. Inductively, we would like to construct a new solution to the size-$(n+1)$ interpolation problem



(2.2.5)

The graphics in the upper left-hand corner is intended to emphasize that we have available to us the previous, order-n augmented solution for the interpolation coefficients in (2.2.3). If we increase the size of this solution by zero-padding, we get

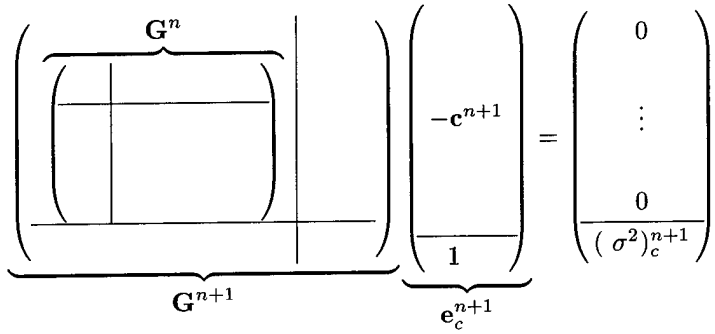


(2.2.6)

with the "leftovers" δ_d^{n+1} given by

$$\delta_d^{n+1} \triangleq G_{n+1,0} - d_1^n G_{n+1,1} - \cdots - d_n^n G_{n+1,n} \tag{2.2.7}$$

We will also make use of the size-$(n+1)$ generalized prediction solution:



(2.2.8)

Since the right-hand side vectors of (2.2.5), (2.2.6), (2.2.8) span a two-dimensional subspace when $(\sigma^2)_c^{n+1}, (\sigma^2)_d^n \neq 0$, we can construct the desired order-$(n+1)$ coefficients $\mathbf{e}_d^{n+1}$ as a linear combination of the two left-hand sides:

$$\mathbf{e}_d^{n+1} = \alpha\,\mathbf{e}_d^n + \beta\,\mathbf{e}_c^{n+1}$$

A little linear algebra then gives the coefficients $\alpha = \alpha^{n+1}$, $\beta = \beta^{n+1}$ for step $n+1$, as indicated below. In the bookkeeping for operation counts, we will assume that the entries $G_{k,l} = \frac{1}{2}\left(r_{k-l} + r_{k+l}\right)$ for the Gramian matrix have been precomputed and stored in advance, with no charge for table-lookup.

Algorithm 2.5 (Interpolation Recursion) *The order-$(n+1)$ interpolation coefficients $g_i^{n+1} = \frac{1}{2}d_i^{n+1}$ can be obtained by the following procedure, which requires $\mathcal{O}(8(n+1))$ steps per cycle:*

1. $$(\sigma^2)_c^{n+1} = G_{n+1,n+1} - \sum_{k=0}^{n} c_k^n\, G_{n+1,k} \tag{2.2.9}$$

2. $$\delta_d^{n+1} = G_{n+1,0} - \sum_{k=1}^{n} (2g_k^n)\, G_{n+1,k} \tag{2.2.10}$$

3. $$\alpha^{n+1} = \frac{(\sigma^2)_c^{n+1}}{(\sigma^2)_c^{n+1} + c_0^{n+1}\delta_d^{n+1}} \tag{2.2.11}$$

4. $$\beta^{n+1} = -\frac{\delta_d^{n+1}}{(\sigma^2)_c^{n+1} + c_0^{n+1}\delta_d^{n+1}} \tag{2.2.12}$$

5. $$g_k^{n+1} = \begin{cases} \alpha^{n+1}\, g_k^n - \frac{1}{2}\,\beta^{n+1} c_k^{n+1}, & 1 \le k \le n \\ \frac{1}{2}\,\beta^{n+1}, & k = n+1 \end{cases} \tag{2.2.13}$$

Initializations: $c_0^0 \leftarrow \frac{G_{1,0}}{G_{0,0}}$, $\delta_d^1 \leftarrow G_{1,0}$, $g_1^1 \leftarrow \frac{G_{1,0}}{G_{1,1}}$

Corollary 2.6 *Recursion for the interpolation error:*

$$(\sigma^2)_g^{n+1} = \alpha^{n+1}\,(\sigma^2)_g^n, \qquad (\sigma^2)_g^1 \leftarrow G_{0,0} - \frac{G_{1,0}^2}{G_{1,1}} \tag{2.2.14}$$

To complete the picture, we will obtain the generalized prediction coefficients c_k^{n+1} from a three-term recurrence.

A Three-Term Recurrence for Generalized Prediction: Three-term recurrences are part of the lore of the orthogonal polynomials (e.g. [5], §3.2); they are widely used in numerical analysis [23], [24] and in describing the special functions of mathematical physics. In signal processing, split algorithms [25] are an example of this sort of procedure. To get an algorithm producing the generalized prediction coefficients c_k^n for the interpolation basis (2.2.1), we start with the Chebyshev-type recursion for each basis element:

$$\boxed{\varphi^{n+1}(z) = 2\,\varphi^1(z)\,\varphi^n(z) - \varphi^{n-1}(z)} \tag{2.2.15}$$

This is just a statement of the trig identity $2\cos\theta\cos k\theta = \cos(k+1)\theta + \cos(k-1)\theta$. It can be used to produce a recursion for the error polynomials e_c^n in (2.2.13):

Algorithm 2.7 (Three-Term Recurrence) *The error polynomial $e_c^{n+1}(z)$ can be generated from the two preceding error polynomials $e_c^n(z)$, $e_c^{n-1}(z)$ in $\mathcal{O}(8n)$ steps from the second-order recursion:*

$$\boxed{e^{n+1}(z) = \left(2\,\varphi^1(z) + B^n\right) e^n(z) + C^n e^{n-1}(z)} \tag{2.2.16}$$

The coefficients in this recursion are calculated via

1. $$(\sigma^2)^n = \langle e^n,\, e^n \rangle = G_{n,n} - \sum_{k=0}^{n-1} c_k^{n-1} G_{n,k} \tag{2.2.17}$$

2. $$B^n = c_{n-1}^{n-1} + \frac{1}{(\sigma^2)^n}\left[\sum_{k=0}^{n-1} c_k^{n-1} G_{n+1,k} - G_{n+1,n}\right] \tag{2.2.18}$$

3. $$C^n = -\frac{(\sigma^2)^n}{(\sigma^2)^{n-1}} \tag{2.2.19}$$

Initializations: $e^0 = \phi^0 = 1$, $(\sigma^2)^0 \leftarrow G_{0,0}$

$$e^1 = \varphi^1 - c_0^0, \quad c_0^0 \leftarrow \frac{G_{10}}{G_{00}}, \quad (\sigma^2)^1 \leftarrow G_{1,1} - \frac{G_{10}^2}{G_{00}}$$

In all these expressions, the elements of the Toeplitz-plus-Hankel Gramian matrix are given by $G_{k,l} = \frac{1}{2}(r_{k-l} + r_{k+l})$.

Sketch of proof: It is easy to see that the coefficient of the highest-order φ vanishes in the function $q(z) = e^{n+1}(z) - 2\varphi^1(z)\,e^n(z)$. So a Gram-Schmidt expansion of $q(z)$ in terms of the preceding error polynomials must look like

$$q(z) = \sum_{k=0}^{n} \frac{\langle q^n,\, e^k\rangle}{\langle e^k,\, e^k\rangle}\, e^k(z) = \sum_{k=0}^{n} \frac{\langle q^n,\, e^k\rangle}{(\sigma^2)^k}\, e^k(z) \tag{2.2.20}$$

Thanks to orthogonality, only the two highest-order dot products on the right are nonzero; these correspond to the coefficients called B^n, C^n.

2.2.6 A Levinson-Type Recursion for Interpolation in Two Dimensions

In this section, we will go through the two-dimensional generalization of the Szegö orthogonalization ideas presented in §2.2.3, with the goal of finding fast methods to solve the normal equations for interpolation. The basic ideas fortunately turn out to be pretty much the same as before; they just require more indices to be expressed. In keeping with the style of previous sections, we will provide a certain amount of representative material in this report, but omit much of the technical detail. In particular, the important definitions regarding orthogonality are made with enough detail to be actually intelligible, but the details about the block structure of a two-dimensional covariance matrix are completely suppressed. For this and the form of the standard Levinson recursion in two dimensions, the reader is referred as usual to [1]. This section concludes by giving just enough detail about a Levinson-type recursion algorithm in two dimensions so that the reader may verify its basic similarity to algorithm 2.4.

Inner Products and Orthogonalization in Two Dimensions: The essential idea for the Szegö orthogonalization process was to express the normal equations in frequency space. To make sense of this in two dimensions, we need to look at the two-dimensional spectral and covariance inner products, which relate expectations in the "time" domain to algebra with z's in the frequency domain. Recall that for a general 2D discrete random process $(x_{k,l})_{k,l}$, the general definition of a **covariance element** is

$$\mathbf{R}_{(i,j),\,(k,l)} = \mathbb{E}\Big\{ (x_{i,j} - \mu_{i,j})\, \overline{(x_{k,l} - \mu_{k,l})} \Big\} \tag{2.2.1}$$

Here, the mean of the process at pixel (i,j) is denoted by "$\mu_{i,j}$." Note that the random process is not assumed stationary in (2.2.1), so the full-blown covariance "matrix" actually requires *four* indices, rather than two. We could make it look more like a familiar one-dimensional covariance by using **multi-index notation**

$$\mathbf{R}_{\mathbf{a},\mathbf{b}} = \mathbb{E}\Big\{ (x_{\mathbf{a}} - \mu_{\mathbf{a}})\, \overline{(x_{\mathbf{b}} - \mu_{\mathbf{b}})} \Big\} = \mathbb{E}\Big\{ (x_{a_1,a_2} - \mu_{a_1,a_2})\, \overline{(x_{b_1,b_2} - \mu_{b_1,b_2})} \Big\} \tag{2.2.2}$$

where $\mathbf{a} \triangleq (a_1, a_2)$ and $\mathbf{b} \triangleq (b_1, b_2)$ are simple, shorthand ways of denoting a vector index, consisting of two component indices.

Recall that for a two-dimensional random process $(x_{k,l})_{k,l}$ to be second-order stationary, the second-order statistics should be shift-invariant:

$$\mathbb{E}\Big\{ (x_{i+p,\,j+q} - \mu_{i+p,\,j+q})\, \overline{(x_{i,j} - \mu_{i,j})} \Big\} = r_{p,q}, \qquad \text{independent of } (i,j) \tag{2.2.3}$$

We will now drop the mean and denote this more briefly as

$$\mathbb{E}\Big\{ x_{i+p,\,j+q}\, \overline{x_{i,j}} \Big\} = r_{p,q} \tag{2.2.4}$$

by replacing the process $(x_{k,l})_{k,l}$ by the process $y_{k,l} \triangleq (x_{k,l})_{k,l} - \mu_{k,l}$ (covariance matrix = correlation matrix). The two-dimensional stationarity property can also be expressed as

$$\mathbf{R}_{(i,j),\,(k,l)} = r_{i-k,\,j-l} \tag{2.2.5}$$

In other words, the stationary covariance array $(r_{k,l})_{k,l}$ involves only two indices, rather than the four needed in (2.2.1), and is an ordinary matrix. The multi-index version of (2.2.5) is simply $\mathbf{R}_{\mathbf{a},\mathbf{b}} = r_{\mathbf{a}-\mathbf{b}}$.

The two-dimensional covariance sequence and the two-dimensional spectral density are related by a two-dimensional $\mathcal{Z}$-transform:

$$(r_{k,l})_{k,l} \quad \overset{\mathcal{Z}}{\longleftrightarrow} \quad S(z_1, z_2) \;=\; \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} r_{k,l}\, z_1^{-k}\, z_2^{-l} \;=\; \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} r_{k,l}\, e^{-ik\theta_1}\, e^{-il\theta_2} \tag{2.2.6}$$

where the second term on the right is obtained by taking z_m to be the unit-circle variable $e^{im\theta}$, $m = 1, 2$. The multi-index version of this fact is denoted

$$(r_{\mathbf{m}})_{\mathbf{m}\in\mathbb{Z}\times\mathbb{Z}} \quad \overset{\mathcal{Z}}{\longleftrightarrow} \quad S(\mathbf{z}) \;=\; \sum_{\mathbf{m}\in\mathbb{Z}\times\mathbb{Z}} r_{\mathbf{m}} \mathbf{z}^{-\mathbf{m}} \;=\; \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} r_{m_1,m_2}\, z_1^{-m_1}\, z_2^{-m_2} \tag{2.2.7}$$

More generally, the multi-index notation

$$\mathbf{x}^{\mathbf{n}} \;\triangleq\; x_1^{n_1}\, x_2^{n_2} \cdots x_p^{n_p} \tag{2.2.8}$$

is useful for denoting polynomials in the variables $x_1, x_2, \ldots, x_p$, i.e. with p-dimensional indices.

Two-Dimensional Spectral Inner Product: Fourier inversion is as straightforward in two dimensions as it is in one dimension, giving a way of recovering the covariance sequence $(r_{k,l})_{k,l}$ from the spectral density $S(\theta_1, \theta_2)$, which is periodic in each variable separately:

$$r_{a,b} \;=\; \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} e^{ia\theta_1}\, e^{ib\theta_2}\, S(\theta_1, \theta_2)\, \tfrac{d\theta_1}{2\pi}\, \tfrac{d\theta_2}{2\pi} \;=\; \oint_{\Gamma_1} \oint_{\Gamma_2} z_1^a\, z_2^b S(z_1, z_2)\, \tfrac{dz_1}{2\pi i z_1}\, \tfrac{dz_2}{2\pi i z_2} \tag{2.2.9}$$

In the second integral on the right, Γ_1 and Γ_2 are simple closed curves enclosing the unit circles $|z_1| = |e^{i\theta_1}| = 1$, respectively $|z_2| = |e^{i\theta_2}| = 1$.

In two dimensions, the basis to orthogonalize for prediction is now the set of all monomials $z_1^{k_1}\, z_2^{k_2}$, where k_1, k_2 are integers. Earlier, we arrived at what the definition of the spectral inner product should be by looking at its behavior for the unit-circle basis, and we can do the same thing in two dimensions. In both cases, this can be done by rewriting the basic Fourier inversion integral above to look more like an inner product of basis functions:

$$\mathbf{R}_{(a,b),\,(k,l)} \;=\; r_{a-k,\,b-l} \;=\; \oint_{\Gamma_1} \oint_{\Gamma_2} z_1^a\, z_2^b\, S(z_1, z_2)\, \overline{z_1^k\, z_2^l}\, \tfrac{dz_1}{2\pi i z_1}\, \tfrac{dz_2}{2\pi i z_2} \tag{2.2.10}$$

The integral on the right will be taken as the definition of the spectral inner product of the basis functions $z_1^a\, z_2^b$ and $z_1^k\, z_2^l$:

$$r_{a-k,\,b-l} \;=\; \left\langle z_1^a\, z_2^b,\, z_1^k\, z_2^l \right\rangle_S \;=\; \left(\mathbf{G}_{z_1,\,z_2}\right)_{(a,b),\,(k,l)} \tag{2.2.11}$$

This is a reminder that the Gramian matrix of inner products of the basis functions depends on four indices, just the way the covariance does in (2.2.1)– this is hardly surprising, considering that the covariance should be equal to this Gramian. In multi-index notation, this can be written

$$r_{\mathbf{m}-\mathbf{n}} \;=\; \left\langle \mathbf{z}^{\mathbf{m}},\, \mathbf{z}^{\mathbf{n}} \right\rangle_S \;=\; \left\langle z_1^{m_1}\, z_2^{m_2},\, z_1^{n_1}\, z_2^{n_2} \right\rangle_S \;=\; \left(\mathbf{G}_{\mathbf{z}}\right)_{\mathbf{m},\mathbf{n}} \tag{2.2.12}$$

The definition of the spectral inner product on basis functions $(z_1^k\, z_2^l)_{k,l}$ can be extended to functions that are square-integrable with respect to the measure

$$d\mu \;=\; S(z_1, z_2)\, \tfrac{dz_1}{2\pi i z_1}\, \tfrac{dz_2}{2\pi i z_2}$$

That is, the **spectral inner product** of $F(z_1, z_2)$ and $G(z_1, z_2)$ is defined to be

$$\left\langle F(z_1, z_2),\, G(z_1, z_2) \right\rangle_S \;=\; \oint_{\Gamma_1} \oint_{\Gamma_2} F(z_1, z_2)\; S(z_1, z_2)\, \overline{G(z_1, z_2)}\, \tfrac{dz_1}{2\pi i z_1}\, \tfrac{dz_2}{2\pi i z_2} \tag{2.2.13}$$

As in one dimension, the spectral inner product in 2D frequency space can be related to a "time-domain" inner product. This is the idea of the 1D generalized Parseval's equality (2.2.5), which we would like to carry through to two dimensions. Rather than speak about the "time domain" as being the inverse of the two-dimensional frequency domain, for 2D images it makes more sense to talk the "space domain" being a transform away from the frequency domain.

In §2.2.3, the one-dimensional spectral inner product $\langle F(z),\, G(z) \rangle_S$ was found to correspond to the inner product of vectors $\langle \mathbf{f},\, \mathbf{g} \rangle_{\mathbf{R}} \;\triangleq\; \mathbf{f}^{\mathrm{T}}\, \mathbf{R}\, \overline{\mathbf{g}}$. Now that we are living in two dimensions, we should replace a singly-indexed vector $\mathbf{f} = \{f_k\}_{k=0}^{n}$ with a doubly-indexed array $\mathbf{f} = \{f_{k,l}\}_{k,l}$, say for $0 \le k \le m$, $0 \le l \le n$. We

will put a arrow above these doubly-indexed vectors here, as a reminder that they are not garden-variety vectors. Likewise, an arrow above the covariance matrix $\vec{\mathbf{R}}$ is a reminder that it is quadruply-subscripted, as in (2.2.1). The **two-dimensional covariance block inner product**, which is a direct generalization of 2.2.4, then can be written

$$\left\langle \vec{\mathbf{f}}, \vec{\mathbf{g}} \right\rangle_{\vec{\mathbf{R}}} = \sum_{\substack{0 \le a \le m \\ 0 \le b \le n}} \sum_{\substack{0 \le k \le m \\ 0 \le l \le n}} f_{a,b} \, \vec{\mathbf{R}}_{(a,b),(k,l)} \, \overline{g_{k,l}} = \sum_{\substack{0 \le a \le m \\ 0 \le b \le n}} \sum_{\substack{0 \le k \le m \\ 0 \le l \le n}} f_{a,b} \, r_{a-k,b-l} \, \overline{g_{k,l}} \tag{2.2.14}$$

where we have used the stationarity property (2.2.5).

In order to derive a two-dimensional Parseval's relation, the two-dimensional polynomial transforms of the multi-vectors $\vec{\mathbf{f}}$, $\vec{\mathbf{g}}$ are needed:

$$\vec{\mathbf{f}} = (f_{a,b})_{\substack{0 \le a \le m \\ 0 \le b \le n}} \quad \overset{\mathcal{Z}}{\longleftrightarrow} \quad \mathcal{P}(f) = \sum_{a=0}^{m} \sum_{b=0}^{n} f_{a,b} \, z_1^a \, z_2^b = F(z_1, z_2) \tag{2.2.15}$$

$$\vec{\mathbf{g}} = (g_{k,l})_{\substack{0 \le k \le m \\ 0 \le l \le n}} \quad \overset{\mathcal{Z}}{\longleftrightarrow} \quad \mathcal{P}(g) = \sum_{k=0}^{m} \sum_{l=0}^{n} g_{k,l} \, z_1^k \, z_2^l = G(z_1, z_2) \tag{2.2.16}$$

Rather than having the sums bristle with indices, we may just write "(k,l)" on the bottom of a sum involving $\vec{\mathbf{g}}$, etc. It is now easy to establish a two-dimensional generalized Parseval relation, based on the fact (2.2.11) that $r_{\mathbf{a},\mathbf{b}} = \langle \mathbf{z}^{\mathbf{a}}, \mathbf{z}^{\mathbf{b}} \rangle$:

$$\begin{aligned} \left\langle \vec{\mathbf{f}}, \vec{\mathbf{g}} \right\rangle_{\vec{\mathbf{R}}} &= \sum_{(a,b)} \sum_{(k,l)} f_{a,b} \, r_{a-k,b-l} \, \overline{g_{k,l}} \\ &= \sum_{(a,b)} \sum_{(k,l)} f_{a,b} \left\langle z_1^a \, z_2^b, \, z_1^k \, z_2^l \right\rangle_S \overline{g_{k,l}} = \left\langle \sum_{(a,b)} f_{a,b} \, z_1^a \, z_2^b, \, \sum_{(k,l)} g_{k,l} z_1^k \, z_2^l \right\rangle_S \\ &= \left\langle F(z_1, z_2), G(z_1, z_2) \right\rangle_S \end{aligned} \tag{2.2.17}$$

We did not even have to write any unit-circle contour integrals out explicitly here: what really counts in the proof is properties of the inner products of basis functions. As a summary, the full statement of the 2D generalized Parseval's relation, including integrals, is

$$\boxed{\begin{aligned} \left\langle \vec{\mathbf{f}}, \vec{\mathbf{g}} \right\rangle_{\vec{\mathbf{R}}} &= \left\langle F(z_1, z_2), G(z_1, z_2) \right\rangle_S \\ \sum_{(a,b)} \sum_{(k,l)} f_{a,b} \, r_{a-k,b-l} \, \overline{g_{k,l}} &= \oint_{\Gamma_1} \oint_{\Gamma_2} F(z_1, z_2) \, S(z_1, z_2) \, \overline{G(z_1, z_2)} \, \tfrac{dz_1}{2\pi i z_1} \, \tfrac{dz_2}{2\pi i z_2} \end{aligned}} \tag{2.2.18}$$

Sketch of Block Recursion for 2D Interpolation: With the usual conventions for block covariances, the recursion for 2D interpolation coefficients can be expressed in terms of block error vectors. For outer and inner interpolation, these are defined respectively by

$$\underline{\mathbf{e}}_f^n = \left(\mathbf{I} \,\middle|\, -\mathbf{F}_{-n}^n \cdots - \mathbf{F}_n^n \,\middle|\, \mathbf{I} \right) = \left(\mathbf{I} \,\middle|\, -\underline{\boldsymbol{f}}^n \,\middle|\, \mathbf{I} \right) \tag{2.2.19}$$

$$\underline{\mathbf{e}}_g^n \overset{\mathcal{Z}}{\longleftrightarrow} = \left(\mathbf{0} \,\middle|\, -\mathbf{G}_{-n}^n \cdots - \mathbf{G}_n^n \,\middle|\, \mathbf{0} \right) = \left(\mathbf{0} \,\middle|\, \underline{\mathbf{e}}_g^n \,\middle|\, \mathbf{0} \right) \tag{2.2.20}$$

$$\underline{\mathbf{e}}_g^{n+1} = \left(-\mathbf{G}_{-n-1}^{n+1} \,\middle|\, -\mathbf{G}_{-n}^{n+1} \cdots - \mathbf{G}_n^{n+1} \,\middle|\, -\mathbf{G}_{n+1}^{n+1} \right) \tag{2.2.21}$$

Underlining is used to indicate that these error vectors are in fact arrays each of whose components is a matrix block.

The algorithm makes use of the solution to outer interpolation, which can be obtained from the standard multichannel Levinson solution by

$$\underline{\boldsymbol{f}}^n = (\underline{\boldsymbol{a}}^{2n+1} \, \underline{\boldsymbol{J}}) + \underline{\boldsymbol{b}}^{2n+1} \tag{2.2.22}$$

Formally, the algorithm is very similar to (2.2.7).

Algorithm 2.8 (Recursion for Interpolation in Two Dimensions)
The order-$(n+1)$ error vector for two-dimensional inner interpolation coefficients can be obtained from the order-n inner interpolation error vector and the multichannel Levinson solution via

$$\underline{\boldsymbol{e}}_g^{n+1} \;=\; \mathbf{C} \underset{\longleftrightarrow}{\underline{\boldsymbol{e}}_g^{n}} + \mathbf{D}\,\underline{\boldsymbol{e}}_f^{n+1} \tag{2.2.23}$$

Step One: Get new $\boldsymbol{\Sigma}, \Delta$, *using coefficients from preceding stage and Levinson solution:*

$$\boldsymbol{\Sigma}_f^n \;=\; 2\,\mathbf{R}_0 + \mathbf{R}_{-2n-2} + \mathbf{R}_{2n+2} \;-\; \underline{\boldsymbol{f}}^n \left[(\underline{\boldsymbol{r}})_+^{2n+1} + (\underline{\boldsymbol{r}})_-^{2n+1}\,\underline{\boldsymbol{J}} \right]^{\mathrm{T}} \tag{2.2.24}$$

$$\boldsymbol{\Delta}_g^{n+1} \;\triangleq\; \boldsymbol{\Delta}_{g,+}^{n+1} + \boldsymbol{\Delta}_{g,-}^{n+1} \;=\; \underline{\boldsymbol{e}}_g^n \left((\underline{\boldsymbol{r}})_+^{2n+1} + (\underline{\boldsymbol{r}})_-^{2n+1}\,\underline{\boldsymbol{J}} \right)^{\mathrm{T}} \tag{2.2.25}$$

Step Two: find new D and C:

$$\mathbf{D} \;=\; \boldsymbol{\Delta}_g^{n+1} \left[\boldsymbol{\Delta}_g^{n+1}\,\mathbf{F}_0^n + \boldsymbol{\Sigma}_f^n \right]^{-1} \;=\; \left[\mathbf{F}_0^n + \boldsymbol{\Delta}_g^{n+1} (\boldsymbol{\Sigma}_f^n)^{-1} \right]^{-1} \tag{2.2.26}$$

$$\mathbf{C} \;=\; (\mathbf{F}_0^n)^{-1}\boldsymbol{\Sigma}_f^n \left[\boldsymbol{\Delta}_g^{n+1} + (\mathbf{F}_0^n)^{-1}\boldsymbol{\Sigma}_f^n \right]^{-1} \;=\; \left[\boldsymbol{\Delta}_g^{n+1} (\boldsymbol{\Sigma}_f^n)^{-1}\mathbf{F}_0^n + \mathbf{I} \right]^{-1} \tag{2.2.27}$$

Having obtained the new, order-$(n+1)$ coefficients $\mathbf{D}(\mathbf{n+1})$ and $\mathbf{C}(\mathbf{n+1})$ from steps one and two, the recursion now produces a new, higher-order error vector from (2.2.24). This block error vector $\underline{\boldsymbol{e}}_g^{n+1}$ contains the new, order-$(n+1)$ block coefficients for the optimum interpolator (= Step Three).

References

[1] M. M. Scheffé. *Interpolation, Orthogonalization and Random Fields.* PhD thesis, Brown University, Division of Engineering, 1996. Available from University Microfilms, 300 N Zeeb Rd, Ann Arbor MI 48103-1500, 1-800-824-0814.

[2] T. Kailath, A. Viera, and M. Morf. Inverses of Toeplitz operators, innovations, and orthogonal polynomials. *SIAM Review*, 20(1):106–119, 1978.

[3] Philippe Delsarte and Yves Genin. On the role of orthogonal polynomials on the unit circle in digital signal processing applications. In Paul Nevai, editor, *Orthogonal Polynomials: Theory and Practice*, pages 115–133. Kluwer Academic Publishers, 1990. (Proceedings of the NATO Advanced Study Institute on Orthogonal Polynomials and Their Applications, Columbus, Ohio, May 22-June 3, 1989).

[4] Thomas Kailath. Notes on the Szegö unit circle orthogonal polynomials in least-squares prediction theory. In Richard Askey, editor, *Gabor Szegö: Collected Papers*, volume I, pages 43–46. Birkhauser, 1992.

[5] Gabor Szegö. *Orthogonal Polynomials.* American Mathematical Society, Providence, R.I., 4th edition, 1975.

[6] L. Ya. Geronimus. *Orthogonal Polynomials.* Consultants Bureau, New York, 1961.

[7] Ulf Grenander and Gabor Szegö. *Toeplitz Forms and Their Applications.* University of California Press, 1958.

[8] T. Kailath. A view of three decades of linear filtering theory. *IEEE Trans. IT*, 20:145–181, 1974.

[9] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 1984.

[10] Theodore S. Chihara. *An Introduction to the Orthogonal Polynomials.* Gordon and Breach, 1978.

[11] Cameron K. Coursey and John A. Stuller. Linear interpolation lattice. *IEEE Transactions on Signal Processing*, 39(4):965–967, 1991.

[12] Gulamabbas A. Merchant and Thomas W. Parks. Efficient solution of a Toeplitz-Plus-Hankel coefficient matrix system of equations. *IEEE Trans. ASSP*, 30(1):40–44, 1982.

[13] S. Lawrence Marple. Fast algorithms for linear prediction and system identification filters with linear phase. *IEEE Trans. ASSP*, 30(6):942–953, 1982.

[14] Andrew E. Yagle. Fast algorithms for structured matrices in signal processing. In N. K. Bose and C. R. Rao, editors, *Handbook of Statistics, Vol. 10: Signal Processing*, pages 933–973. Elsevier, 1993.

[15] Benjamin Friedlander and Martin Morf. Least squares algorithms for adaptive linear-phase filtering. *IEEE Trans. ASSP*, 30(3):381–389, 1982.

[16] Georg Heinig, Peter Jankowski, and Karla Rost. Fast inversion algorithms of Toeplitz-plus-Hankel matrices. *Numer. Math*, 52:665–682, 1988.

[17] Jin-Jen Hsue and Andrew E. Yagle. Fast algorithms for close-to-Toeplitz-plus-Hankel systems and two-sided linear prediction. *IEEE Transactions on Signal Processing*, 41(7):2349–2361, 1993.

[18] Steven M. Kay. Some results in linear interpolation theory. *IEEE Trans. ASSP*, 31:746–749, 31.

[19] Ali H. Sayed, Hanoch Lev-Ari, and Thomas Kailath. Fast triangular factorization of the sum of quasi-Toeplitz and quasi-Hankel matrices. *Linear Algebra and Its Applications*, 191:77–106, 1993.

[20] Christopher J. Zarowski. Schur algorithms for Hermitian Toeplitz, and Hankel matrices with singular leading principal submatrices. *IEEE Transactions on Signal Processing*, 39(11):2464–2480, 1991.

[21] Christopher J. Zarowski. A Schur algorithm and linearly connected processor array for Toeplitz-plus-Hankel matrices. *IEEE Transactions on Signal Processing*, 40(8):2065–2078, 1992.

[22] William F. Trench. An algorithm for the inversion of finite Hankel matrices. *J. Soc. Indust. Appl. Math.*, 13(4):1102–1107, 1965.

[23] Walter Gautschi. Computational aspects of three-term recurrence relations. *SIAM Review*, 9(1):24–82, 1967.

[24] Walter Gautschi. A survey of Gauss-Christoffel quadrature formulas. In P. L. Butzer and F. Fehér, editors, *E. B. Christoffel: The Influence of His Work on Mathematics and the Physical Sciences*, pages 72–147. Birkhauser, 1981.

[25] Philippe Delsarte and Yves Genin. A survey of the split approach based techniques in digital signal processing applications. *Phillips J. Res.*, 43:346–374, 1988.